UNIVERSITY OF LIVERPOOL

Person Re-identification and Tracking in Video Surveillance

A thesis submitted in accordance with the requirements of the University of Liverpool
for the degree of Doctor in Philosophy
by

Yanchun Xie

Department of Electrical Engineering and Electronics
School of Electrical Engineering and Electronics and Computer Science
University of Liverpool

June, 2020

Abstract

Video surveillance system is one of the most essential topics in the computer vision field. As the rapid and continuous increasement of using video surveillance cameras to obtain portrait information in scenes, it becomes a very important system for security and criminal investigations. Video surveillance system includes many key technologies, including the object recognition, the object localization, the object re-identification, object tracking, and by which the system can be used to identify or suspect the movements of the objects and persons. In recent years, person re-identification and visual object tracking have become hot research directions in the computer vision field. The re-identification system aims to recognize and identify the target of the required attributes, and the tracking system aims at following and predicting the movement of the target after the identification process.

Researchers have used deep learning and computer vision technologies to significantly improve the performance of person re-identification. However, the study of person re-identification is still challenging due to complex application environments such as lightning variations, complex background transformations, low-resolution images, occlusions, and a similar dressing of different pedestrians. The challenge of this task also comes from unavailable bounding boxes for pedestrians, and the need to search for the person over the whole gallery images.

To address these critical issues in modern person identification applications, we propose an algorithm that can accurately localize persons by learning to minimize intra-person feature variations. We build our model upon the state-of-the-art object detection framework, i.e., faster R-CNN, so that high-quality region proposals for pedestrians can be produced in an online manner. In addition, to relieve the negative effects caused by varying visual appearances of the same individual, we introduce a novel center loss that can increase the intra-class compactness of feature representations. The engaged center loss encourages persons with the same identity to have similar feature characteristics.

Besides the localization of a single person, we explore a more general visual object tracking problem. The main task of the visual object tracking is to predict the location and size of the tracking target accurately and reliably in subsequent image sequences when the

target is given at the beginning of the sequence. A visual object tracking algorithm with high accuracy, good stability, and fast inference speed is necessary. In this thesis, we study the updating problem for two kinds of tracking algorithms among the mainstream tracking approaches, and improve the robustness and accuracy.

Firstly, we extend the siamese tracker with a model updating mechanism to improve their tracking robustness. A siamese tracker uses a deep convolutional neural network to obtain features and compares the new frame features with the target features in the first frame. The candidate region with the highest similarity score is considered as the tracking result. However, these kinds of trackers are not robust against large target variation due to the no-update matching strategy during the whole tracking process. To combat this defect, we propose an ensemble siamese tracker, where the final similarity score is also affected by the similarity with tracking results in recent frames instead of solely considering the first frame. Tracking results in recent frames are used to adjust the model for a continuous target change. Meanwhile, we combine adaptive candidate sampling strategy and large displacement optical flow method to improve its performance further.

Secondly, we investigate the classic correlation filter based tracking algorithm and propose to provide a better model selection strategy by reinforcement learning. Correlation filter has been proven to be a useful tool for a number of approaches in visual tracking, particularly for seeking a good balance between tracking accuracy and speed. However, correlation filter based models are susceptible to wrong updates stemming from inaccurate tracking results. To date, little effort has been devoted to handling the correlation filter update problem. In our approach, we update and maintain multiple correlation filter models in parallel, and we use deep reinforcement learning for the selection of an optimal correlation filter model among them. To facilitate the decision process efficiently, we propose a decision-net to deal with target appearance modeling, which is trained through hundreds of challenging videos using proximal policy optimization and a lightweight learning network. An exhaustive evaluation of the proposed approach on the OTB100 and OTB2013 benchmarks show the effectiveness of our approach.

Key Words: Object Tracking, Correlation Filter, Feature Learning, Reinforcement Learning, Re-identification, Person search, Center loss

Acknowledgement

Firstly, I would like to express my sincere gratitude to my advisor Prof. Jimin Xiao for the continuous support of my Ph.D study and related research, for his patience, motivation, and immense knowledge. His guidance helped me in all the time of research and writing of papers including this thesis. I could not have imagined having a better advisor and mentor for my Ph.D study. I would like to thank Prof. Tammam TILLO, Prof. Kaizhu Huang, Prof. Mark Leach and Dr. Chao YAO, especially my IPAP advisors Prof. Qiufeng Wang and Prof. Al-Nuaimy Waleed, for their insightful comments and encouragement, but also for the hard questions which incented me to widen my research from various perspectives.

I would like to express my sincere thanks to the University AI research center in XJTLU and Prof. Eng Gee Lim. For supporting me and giving the opportunity to work on the Face Recognition project in the last year.

My sincere thanks also goes to my labmates: Dr. Zhi Jin, Dr. Fei Cheng, Dr. Li Yu, Dr. Samer Jammal, Dr. Haochuan Jiang, Mr. Boyuan Sun, Mr. Shufei Zhang. Mr. Tianhong Dai, Mr. Dingyuan Zheng, Mr. Zhuang Qian, Mr. Bingfeng Zhang, Mr. Mingjie Sun, Mrs. Hui Li, Mrs. Siyue Yu. Without they precious support it would not be possible to conduct this research. And also I treasure the days being with all my friends in our Lab MMT408.

I thank my fellow labmates in for the stimulating discussions, for the sleepless nights we were working together before deadlines, and for all the fun we have had in the last four years.

Last but not the least, I would like to thank my parents and girl friend for supporting me spiritually throughout writing this thesis. Without their support, and love during the past few years, I could not finish this thesis.

Contents

List of Figures

List of Tables

List of Algorithms

Chapter 1

Introduction

1.1 Research Motivation

Video surveillance is an important security application in public areas. Among the components of video surveillance systems, pedestrian tracking and retrieval are the primary problems of monitoring. The person re-identification technology is an important guarantee for pedestrian tracking and retrieval. Person re-identification and object visual tracking are essential components of computer vision and video processing. Great efforts have been made by researchers in the past decades, while there still exists many challenging problems like occlusions, low image quality, and motion blur to be studied in both person re-identification and visual object tracking tasks. Due to the different settings between the re-identification and the tracking tasks, so in this thesis, we study the person re-identification problem and the object tracking problem separately.

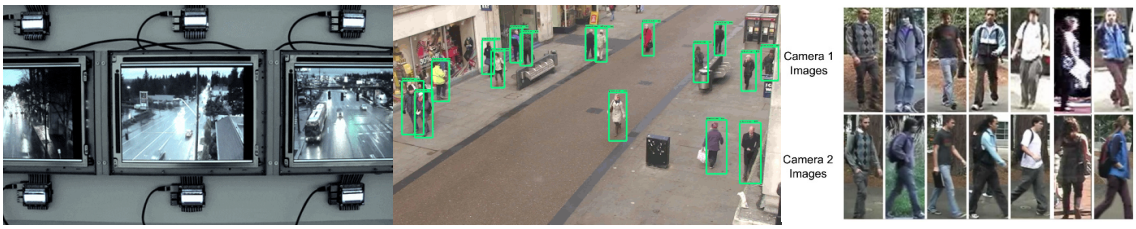


Fig. 1.1: Application of the person re-identification in public space.

As shown in Fig. 1.1. In the industry, given a large and open public space with multiple surveillance cameras, it is difficult to find certain people under great populations of distractors and heavy occlusions. Traditionally people use face recognition technology to identify persons. However, under such an environment, faces are usually small and suffer from motion blur and occlusion, and the faces do not always appear among the captured frames.

Person re-identification is to re-identify the same person across different cameras,

which has attracted increasingly more interest in recent years [5, 6, 7, 8, 9, 10, 11]. The emergence of this task is mainly stimulated by the increasing demand for public security and widespread surveillance camera networks among public places, such as airports, universities, shopping malls, etc. The obtained images from surveillance cameras are usually with some characteristics, e.g., low-quality, variable, and contain motion blur. Traditional biometrics, such as the face, iris, and fingerprint, are generally not available. Thus, many person re-identification applications exploit reliable body appearance.

In reality, perfect pedestrian bounding boxes are unavailable in surveillance scenarios. Besides, existing pedestrian detectors unavoidably produce false alarms, misdetections, and misalignments. All these factors compromise the re-identification performance. Therefore, current re-identification algorithms cannot be directly applied to real surveillance systems, where we need to search for a person from whole images.

While the majority of existing person re-identification works engage boxes manually annotated or produced by a fixed detector in their applications. Based on the observation, we must study the impact of pedestrian detectors on re-identification accuracy. Specifically, [12, 13] showed that considering detection and re-identification jointly leads to higher person search accuracy than optimizing them separately. Thereby, the detector and re-identification parts can interact with each other so as to reduce the influence of detection misalignments.

Meanwhile, the feature learning in re-identification is vital. Early works show that such kind of identification task could greatly benefit from the feature learning [14]. It is found that the identification task increases the inter-personal variations by drawing features extracted from different identities apart, while the verification task reduces the intra-personal variations by pulling features extracted from the same identity together [15].

Visual object tracking is also another important components in video surveillance. The primary task of visual object tracking is to estimate the trajectory of a target in a video sequence. The illumination variation, occlusion, rotation, camera motion, and deformation are still the challenges for visual object tracking tasks.

The model updating in visual object tracking remains to be a problem which can be found in every kind of trackers. Recently, there are two main approaches in the visual tracking community: the siamese trackers and the correlation filter based trackers.

Siamese instance search tracker [16] is one of the classic siamese tracking algorithms, which proposes an ideal matching function for visual tracking task. The goal of SINT [16] is to learn a generically applicable matching function from the annotated video dataset, which is sufficiently large to model the invariance factors of different videos. Once the matching function has been trained on the external video dataset, the matching function

will not be updated anymore during the tracking process. The drawbacks of the approach is that it has no model updating, no combination of different tracking algorithms and no occlusion detection. This kind of tracker simply returns the candidate region in a new frame that has the highest similarity score with the initial target in the first frame. Nevertheless, with such a simple model, experimental results point out that the tracker is robust to handle the common variation of targets. Meanwhile, the matching function can be used to track unseen targets without being updated while leading to a comparable performance with existing tracking methods. So we consider that a good tracker should be mitigating this problem. In our thesis, we propose an ensemble siamese Tracker, where the final similarity score is also affected by the similarity with tracking results in recent frames instead of solely considering the first frame to improve the robustness of the siamese tracker further.

In order to obtain discriminative features for tracking, owing to the underlying complexity of parameter models, significant amount of computational resources are needed. In addition to this, large models tend to introduce severe over-fitting problems. Models like VGG-19 tend to be an inferior option for CF-based trackers. Other than one forward pass in the convolutional network for feature extraction, CF trackers need additional time to compute the correction filter in the Fourier frequency domain which can hardly benefit from GPUs. Nevertheless, operating in the Fourier frequency domain speeds up CF. In this thesis, we also study the feature extractor in correlation filter based method, and introduce a light-weight feature extractor so that our approach can be deployed in real-time applications, where the frame rates are high.

Moreover, for the correlation filter based trackers, most discriminative model-based trackers exploit the target from a given bounding box directly, which is used to build the appearance model of the objects at the latter stages. During the tracking process, new image patches generated from new frames are supplemented to update the CF model further. Generally, a small update-rate is usually preferred for CF trackers in order to maintain model stability. These trackers may easily suffer from a drift problem, especially in challenging environments such as partial occlusions, background clutter, and low resolution. Based on these observations, we studied the updating problem in the correlation filter based object tracking and propose to use reinforcement learning to assist in the decision making process in the CF model update.

1.2 Person Re-identification

CNN-based deep learning models have attracted much attention and been successfully applied on person re-identification since two pioneer works [17, 18, 5, 6, 7, 8, 9, 10, 11].

It is a fundamental task in surveillance systems and has widespread application prospects in numerous fields.

A person re-identification task is defined as follows: Given an image of a pedestrian captured from one camera, try to identify this person from the gallery set captured by other different cameras. It is a challenging problem since the appearance of pedestrians can change significantly between different cameras. The emergence of this task is mainly due to the increasing demand for public security and an extensive network of surveillance cameras in public places, such as airports, universities, shopping malls, etc. The images obtained from surveillance cameras are usually with common characteristics, e.g., low-quality, variable, and contain motion blur. Traditional biometrics, such as the face, iris, and fingerprint, are generally not available. Thus, most person re-identification applications exploit reliable body features.

Generally, two categories of CNN models are commonly employed in this community. One category is the representation learning-based classification model as used in image classification and object detection. The other category is the metric learning-based siamese model using image pairs [17, 19] or triplets [9] as input.

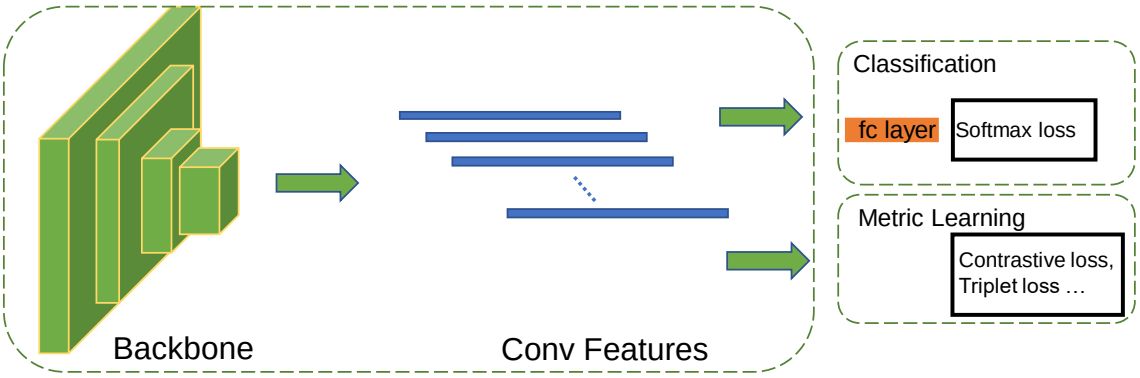


Fig. 1.2: An illustration of the person re-identification framework. The netwok use a pretrained backbone for feature extraction, both classification loss and metric learning loss can be used in the training.

Representation learning is a very common method for person re-identification. Although the goal of the person re-identification task is to learn the similarity between two pictures, these methods do not directly consider the similarity between pictures when training the network. The methods treat the person re-identification task as a classification problem or a verification problem. The characteristic of this method is that the last layer of the network is not a feature vector of the input image but a classification result after the softmax activation function. The methods consider the person re-identification as a classification problem using the persons' identification or attributes as training labels to train the classification model, and the training only needs single image; While for the

verification problem, a pair of pictures are sent into the model, and the network would learn both whether the pictures belong to the same person.

Suppose the training set has n pictures of K pedestrians, and denotes the input picture x and the network is f. The last layer of the network outputs the prediction vector $z = [z_1, z_2, \cdots, z_k] \in \mathbf{R}^K$. Therefore, picture x belongs to $k(k \in 1, 2, 3, \cdots, K)$. The probability of the pedestrian ID is. So the classification loss of the network is:

$$L_{softmax}(f, x) = -\sum_{k=1}^{K} q(k) \log p(k), \tag{1.1}$$

where q is the label of x.

Metric learning is a method widely used in the field of image retrieval. Different from the representation learning, metric learning aims to learn the similarity of two images through the models. For the person re-identification problem, the similarity between different persons' pictures should be less than that of the pictures of the same person. Specifically, we can define a mapping function $f(\cdot)$, to map the original image from the image domain to the feature domain, and then use a distance metric function to calculate the distance between the feature vectors. Finally, by minimizing the loss of the network, find an optimal mapping $F(\cdot)$, so that the distance between two pictures of the same person is as small as possible. This mapping function is a trainable deep neural convolutional network with a siamese structure.

Contrastive loss is a widely used metric loss. The input of the twin network is a pair of pictures I_a and I_b. These two pictures can be the same person or different persons. Each pair of training pictures has a label y, where y = 1 means that the two pictures belong to the same one (positive sample pair), and y = 0 means that they belong to different person(negative sample pair). Then, the contrastive loss function is defined by:

$$L_{contrast} = y d^2_{l_a, I_b} + (1 - y) max\left(0, \alpha - d^2_{I_a, I_b}\right), \tag{1.2}$$

where d is the feature distance in the feature space, which can be the Euclidean distance or the Cosine distance, and α is the threshold parameter.

Most re-identification datasets provide two images for each pedestrian such as VIPeR [20], CUKH01 [21] CUHK03 [18], therefore, currently most CNN-based re-identifications schemes use the Siamese model. In [17], an input image is partitioned into three overlapping horizontal parts, and the parts go through two convolutional layers plus one fully connected layer, which fuses them and outputs a vector to represent this image, and lastly, two vectors are connected by a cosine layer. Ahmed et al. [19] improved the Siamese model by computing the cross-input neighborhood difference features, which compared the features from one input image to features in neighboring locations of the other image. In [22], Varior et al. incorporated long short-term memory (LSTM) modules into

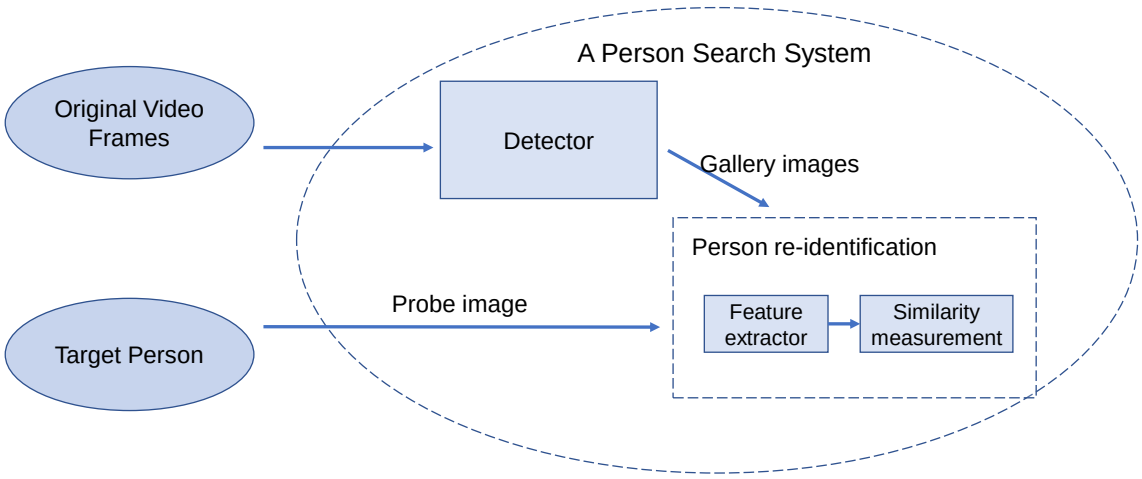


Fig. 1.3: The person search system. The person search problem is divided into two separate tasks: pedestrian detection and person re-identification.

a siamese network so that the spatial connections can be memorized to enhance the discriminative ability of the deep features. Similarly, Liu et al. [23] proposed to integrate a soft attention-based model in a siamese network to adaptively focus on the crucial local parts of the input image pair.

One disadvantage of the siamese model is that it cannot take full advantage of the re-identification annotations. The Siamese model only considers pairwise labels (similar or not similar), which is a weak label. Another potentially effective strategy is to use a classification/identification mode, which makes full use of the re-identification labels. On large datasets, such as PRW and MARS, the classification model achieves excellent performance without careful pairwise or triplet selection [24, 13].

Technically, a person re-identification system for video surveillance consists of three components, including person detection, person tracking, and person retrieval. While the first two components are independent computer vision tasks, most person re-identification studies focus on the third component. Numerous re-identification algorithms as well as datasets[17, 18, 25, 26, 27, 28] have been proposed during the past decades, and the performance on these benchmarks has been improved substantially. All these algorithms focus on the third component of the pipeline, assuming the person/pedestrian detection is already available. In other words, a query person is matched with cropped pedestrians in the gallery instead of searching for the target person from whole images. In reality, perfect pedestrian bounding boxes are unavailable in surveillance scenarios. Besides, existing pedestrian detectors unavoidably produce false alarms, misdetections, and misalignments.

We divide the person search problem into two separate tasks: pedestrian detection and person re-identification. As shown in Fig. 1.3. Most re-identification datasets provide only two images for each pedestrian such as VIPeR [20], CUKH01 [21], CUHK03 [18],

(a) Person re-id: matching with manually cropped pedestrians

(b) Person search: Finding from whole scene images

Fig. 1.4: The difference of person reid and person search.[2]

therefore, currently most CNN-based re-identification schemes usuall use the Siamese model. Pedestrian detectors DPM [7], ACF [6], and Checkerboards [41] are the most commonly used off-the-shelf detectors. They use hand-crafted features and linear classifiers to detect pedestrians. In recent years, many factors, such as CNN model structures, training data, and different training strategies, have been studied. CNN based pedestrian detectors have also been developed [29, 30, 31].

Different from previous methods, we jointly handle both aspects in a Single Convolutional Neural Network to study the impact of pedestrian detectors. The difference of person reid and person search is shown in Fig. 1.4. Our proposed method consists of two parts, given a whole input gallery image, a pedestrian proposal net is used to produce bounding boxes of candidate people, which are fed into an identification net to extract features for comparing with the target person. The pedestrian proposal net and the identification net adapt to each other during the joint optimization. To the best of our knowledge, end-to-end deep learning for person search [1] is the first work to introduce an end-to-end deep learning framework to handle the challenges from both detection and re-identification jointly. Thereby, the detector and re-identification parts can interact with each other so as to reduce the influence of detection misalignments. Meanwhile, misalignments of proposals are also acceptable, as the identification net can further adjust them.

Since the pedestrians appearing in each image are random and unbalanced, we can-

not directly introduce such tasks into the faster R-CNN framework. It is challenging to organize an equivalent amount of positive and negative pedestrian pairs for an enormous amount of identities. To relieve the negative effect caused by various visual appearances of the same individual, we introduce the center loss [32] that can increase the intra-class compactness of feature representations is introduced. The center loss encourages learned pedestrian representations from the same class to share similar feature characteristics. Moreover, our proposed module can be embedded in any CNN-based person search framework for improving performance.

1.3 Overview of Visual Object Tracking

Tracking is used to localize the person which has been identified in the video surveillance system. And also visual tracking is one of the most fundamental problems in the field of computer vision. It is a task that locates target objects precisely over a sequence of frames only given a bounding box in the first frame. Visual tracking has a wide range of practical applications, such as video surveillance, autonomous driving, and robotics. It acts as an essential component in many other computer vision tasks. In video surveillance systems, the tracking algorithm can provide realtime location information with fast speed. Autonomous driving systems need to obtain locations of different vehicles and persons for their safety systems. With the rapid development of high-end smartphones, a stable face tracking algorithm has served as a premise for further face analysis. In robotics tasks, the decision making and navigation relies highly on the imformation and is features captureed and extracted by the tracking of objects.

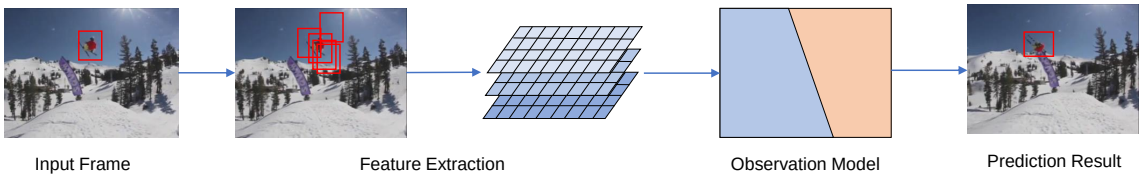


Fig. 1.5: The standard pipline of a visual object tracking framework.

The traditional tracking algorithm aims at predicting the moving state of the target, while the visual tracking now in computer vision is a more complex system that consists of motion model, feature extraction, observation model, model update, and other integration strategies, a stand pipline of a visual object tracking is shown in Fig 1.5.

(1) Motion model: Generate a large number of candidate samples.

(2) Feature extraction: Extract features to represent the target.

(3) Observation model: Score among candidate samples.

(4) Model update: Update the observation model to adapt to the appearance changes of the target.

(5) Integration method: Integrate multiple decisions to obtain a better decision result.

The five research contents are briefly introduced below.

Motion Model: The speed and accuracy of generating candidate samples directly determine the performance of the tracking algorithm. There are two commonly used methods: Particle Filter and Sliding Window. Particle filtering is a Bayesian sequential estimation, which uses a recursive method to model the hidden state of the target. The sliding window is a dense sampling method, which lists all possible samples as candidate samples around the target.

Feature Extractor: Discriminative feature representation is one of the keys to target tracking. Commonly used features are divided into two types: hand-crafted features and deep features. Hand-designed features like gray features, directional histograms (HOG), Haar-like, and scale-invariant features (SIFT) are commonly used. Unlike artificially designed features, deep features are features learned through a large number of training samples, which are more discriminative than manually designed features. Therefore, tracking methods using deep features usually perform better.

Observation Model: The existing visual tracking method can be typically categorized into two kinds: Generative method and Discriminative method. Benefitted from the rising improvement of deep learning, the latter methods have become the mainstream method in visual object tracking.

Generative methods focus on computing the similarity between the tracking object and targets by learning a possibility model. Generative methods firstly extract the target features to learn the appearance model representing the target. Then use the generated possibility model to search the image area, and the area that best matches the model in the image is the target location. The possibility models are built on target features. Thus the step of extracting features for the tracking object is vital. Many research works have been proposed to find a better describe model, including template model [33, 34], Gaussians Mixture Model [35], subspace model [36] etc. After modeling the description models, the similarity matching process can be applied in the new search region to find the match tracking target. Generative tracking methods can obtain more image information and will produce more accurate results in complex environments. However, these methods ignore context information. Model drift often occurs when distractors appear, and the model is vulnerable to interference.

While the discriminative methods usually consider tracking as a classification problem. Despite the generative methods, the discriminative methods study the prediction model without considering the generation model of the tracking target. Usually, a classifier is constructed and trained by a machine learning optimizer to distinguish the background from the tracking target. Most classic machine learning techniques have been attempted in the visual tracking field, such as support vector machine, Bayes, multiple instance learning, metric learning [37, 38, 39, 40]. In recent years, two main kinds of tracking methods have been developed and been investigated deeply in the visual tracking community: one is the correlation filter based method, the other is the deep learning based method. Correlation filter methods often classify by computing reliable confidence scores in a dense 2D Gauss distribution. These methods perform fast training and inference by transforming the objective function into the Fourier domain. The Deep Learning methods make better use of training data, and the deep feature descriptors can improve the

models'ability to distinguish the target from its neighborhood background. The learning methods highly rely on the training datasets. However, for the visual object tracking task, the target location is only given in the first frame, the lack of training data brings great challenges.

Model Update: The model update is mainly to update the observation model to adapt to the apparent change of the target and prevent the tracking process from drifting. The lack of positive and negative labels in visual tracking limits the ability of the observation model. One solution to this circumstance is to use prior knowledge such as VGG networks, which are pretrained on the large scale image dataset. Discriminative correlation filter based trackers have mainly adopted deep features to develop an advanced observation model, while it is another challenge on online learning during tracking. Usually, the correlation filter uses a moving average to update their trained models, and deep learning based models either not update the appearance template like siamese methods or fine-turn the classifier during the tracking process to meet. In [41], Nam et al. combine pretrained convolutional layers and multiple fully connected layers for specific sequences to achieve better tracking performance. Most methods choose the offline training and online finetuning strategy such as SO-DLT [42] and MDNet[41]. SANet[43]introduces an additional recurrent neural network structure to enhance object representation. In this thesis, different form above deep learning based methods, we propose a feature ensemble method and an unsupervised model selection through reinforcement learning to handle the model update problem in visual object tracking.

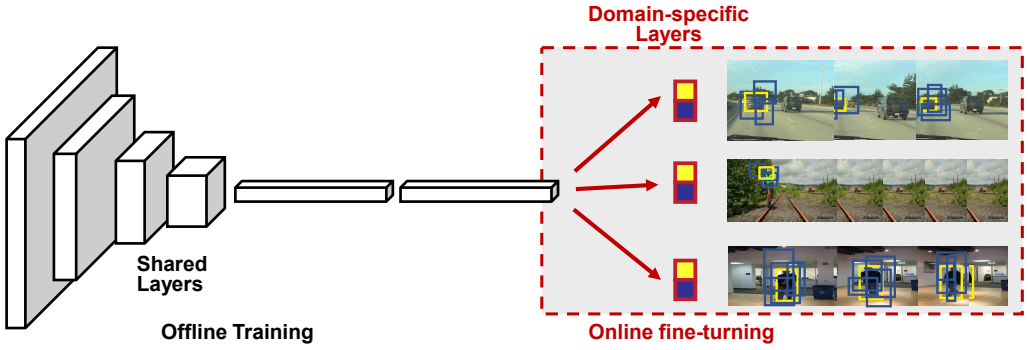


Fig. 1.6: The illustrate figure of a offline training and online fineturning framework in visual object tracking.[41]

Integration method: The integration method is helpful in improving the prediction accuracy of the model. It is often regarded as an effective way to improve tracking accuracy. The integration method can be divided into two categories in general: pick the best among multiple prediction results, or use the weighted average of all predictions.

1.4 Deep Visual Tracking

1.4.1 Developments by deep learning

In this section, we will introduce the development of DeepLearning Trackers and detail the algorithm that been used in our proposed methods.

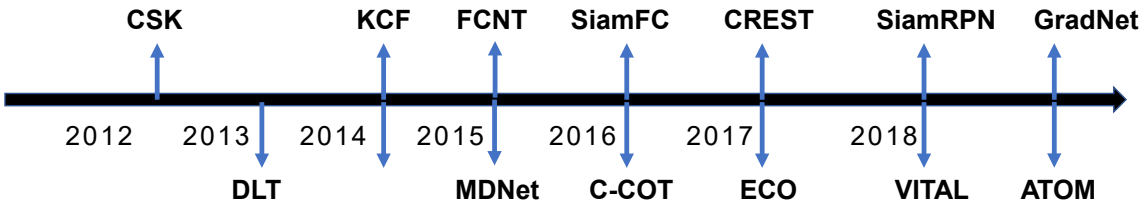


Fig. 1.7: The major development of visual object tracking since 2012. CSK[44], DLT[45], KCF[46], FCNT[47], MDNet[41], SiamFC[48], C-COT[49], ECO[50], SiamRPN[51], VITAL[52], ATOM[53], GradNet[54].

Most computer vision tasks such as classification and object detection have benefited from convolutional neural networks in recent years, as well as visual object tracking. Firstly, researchers explore and study deep features to exploit traditional methods [47]. Offline training or finetuning of deep neural networks like Alexnet and VGG is adopted for visual tracking purposes [41]. In the year 2016, the siamese network [48] is proposed for realtime tracking, and it becomes a trend to integrate deep neural networks into traditional frameworks. After that, temporal and contextual information is further studied in the community, and also researchers attempt to train the network on various large-scale video datasets [55, 56]. At the same time, researches study the influence of different learning and search strategies and try to design more sophisticated architectures [57] for visual tracking tasks. In the following, reinforcement learning based and GAN based methods are presented. In recent research, the deep detection and segmentation approach [58, 59] is investigated upon the siamese structure for visual tracking.

The siamese network-based methods balance the performance and speed of object tracking. Therefore its structure attracts the most attention of researchers in the field. The siamese tracker tracking by similarity comparison. The tracker searches for the candidate most similar to the exemplar given in the start frame, by a learned prior deep siamese similarity function. The siamese branch consists of deep neural networks that take advantage of deep learning. During the tracking phase, the branch remains fixed, and the appearance model does not update. The siamese structure produces a promising result with a fast running speed on public tracking benchmarks. In the next session, we will introduce two of the classical siamese trackers.

1.4.2 Siamese Trackers

Recently, the Siamese network based trackers have received significant attention for their well-balanced tracking accuracy and efficiency. These trackers formulate visual tracking as a cross-correlation problem and are expected better to leverage the merits of deep networks from end-to-end learning. Two typical siamese networks in tracking will be introduced.

Tao et al. [16] first propose a siamese instance search for tracking. Instance search from one example, also known as particular object retrieval, is related to object tracking. The most popular exemplification is based on matching local image descriptors between the query and the candidate image. In order to learn a robust representation for instance search of less textured, more generic objects, SINT propose to learn a robust matching function for matching arbitrary, generic objects that may undergo all sorts of appearance variations. Instead of focusing on a specific category person or vehicle and learning from a clean dataset, the method introduces a universal matching model that is suited for tracking that applies to all categories and all realistic imaging conditions.

This approach operates on pairs of data in order to study a matching function to adapt objects' appearance change. The network structure builds on top of convolutional neural networks with two branches. The network takes the two inputs separately while sharing the same parameters. Only a few max pooling layers are used for the sake of precise localization. To reduce the computation, which costs by evaluating hundreds of candidate regions for the new frame, the region pooling layer is used for the fast processing of multiple overlapping regions. Each branch of the network takes as input one image and a set of regions. The region pooling layer converts the feature map from a particular region into a fixed-length representation. In the end, the two branches in the siamese instance network are connected with a single loss layer, where

$$\mathcal{L}\left(x_j, x_k, y_{jk}\right) = \frac{1}{2} y_{jk} D^2 + \frac{1}{2}\left(1 - y_{jk}\right) \max\left(0, \epsilon - D^2\right) \tag{1.3}$$

where $D = \left\| f\left(x_j\right) - f\left(x_k\right) \right\|_2$ is the Euclidean distance of two ℓ_2 -normalized latent representations, $y_{jk} \in \{0, 1\}$ indicates whether x_j and x_k are the same object or not, and ϵ is the minimum distance margin that pairs depicting different objects should satisfy.

In the tracking inference, once the matching function is learned, we can pass all the candidate boxes from the search image and pick the candidate box that matches best to the original target in the first frame,

$$\hat{x}_t = \underset{x_{j,t}}{\arg\max} m\left(x_{t=0}, x_{j,t}\right) \tag{1.4}$$

where $x_{j,t}$ are all the candidate boxes at frame t, m is the learned matching function, $m(x, y) = f(x)^T f(y)$

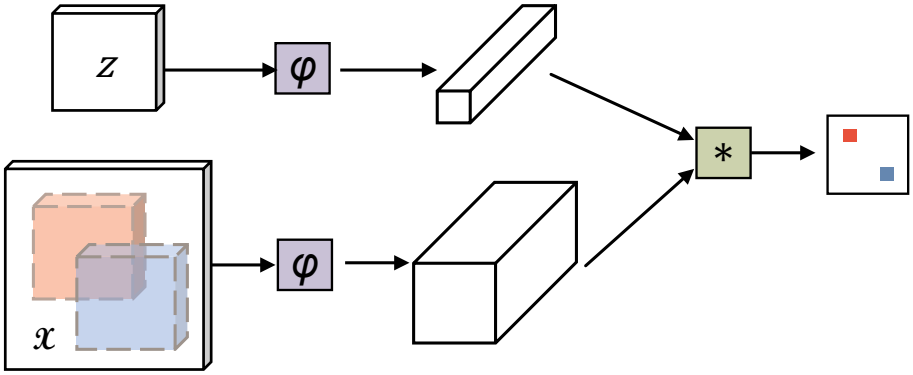


Fig. 1.8: The structure of a siamese framework for visual object tracking[48].

At the same time, Luca et al. [48] propose a similar learning-based algorithm for object visual tracking. They propose to learn a function that compares an exemplar image to a candidate image of the same size and returns a high score if the two images depict the same object and otherwise a low score.

Due to the deep neural networks which trained from large scale supervised dataset have been widely adopted in many other computer vision tasks, Luca et al. attempt to make full use of the large data resources. Another obstacle in visual tracking is that since any arbitrary object may become the tracking target, it is not possible for a tracker to train a specific detector ahead. The constraint of realtime in tracking tasks makes it hard to apply object detection on each sequence of frames directly.

In the method of [48], all possible locations will be tested to find the position of the object in new frames. The candidate with the maximum similarity to the past appearance of the object will be selected as the target location. The similarity function will be learned from a dataset of videos with labeled object bounding boxes.

Specifically, with respect to the search image x, which contains multiple candidates, a fully-convolutional function is designed to commutes with translation in [48]. As shown in Fig. 1.8. Denote the translation operator $(L_\tau x)[u] = x[u - \tau]$, a function h is fully-convolutional with integer stride k if

$$h(L_{k\tau}x) = L_\tau h(x), \tag{1.5}$$

for any translation τ.

Instead of a candidate image of the same size, this method can send a much larger search image to the network, and it will compute the similarity at all translated subwindows on a dense grid in a single forward process. The output of this network is a score map defined on a finite grid $D \subset \mathcal{Z}^2$. During tracking, a search image centered on the previous position of the target will be used. Position of the highest score relative to the center. The position of the maximum score represents the center of the score map, mul-

tiplied by the stride of the network, which gives the displacement of the target from one frame to the next. Multiple scales can be handled by assembling a batch of scaled images.

1.4.3 Correlation Filter Trackers

Historical Developments

CF-based trackers achieve a good trade-off between accuracy and speed by solving a ridge regression problem efficiently in the Fourier frequency domain. After Bolme *et al.* introduced the CF for fast visual tracking, several bodies of work have been proposed to improve the tracking performance of CF-based approaches. Henriques *et al.* propose a circulant structure kernel tracker (CSK) [44]. A high-speed tracker with kernelized correlation filters (KCF) is proposed in [46]. In KCF [46], a multi-channel Histogram Of Gradient (HOG) feature is introduced to calculate the CF. Danelljan *et al.* introduces a scale pyramid representation [60] to handle the scaling issue and proposed the 3-dimensional CF. In [61], separate discriminative correlation filters were learned for translation and scale estimation, respectively. To mitigate unwanted boundary effects, Danelljan *et al.* introduced a spatially regularized term [62] to penalize CF coefficients based on their spatial locations. Unfortunately, the improvement in accuracy goes along with significant reductions in tracking speed.

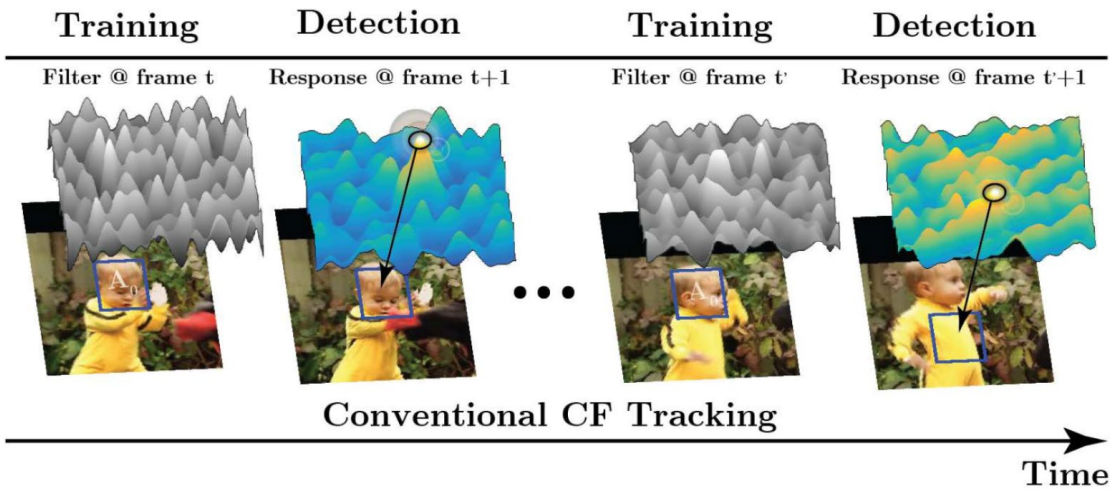


Fig. 1.9: The standard pipline of a CF-based tracking framework[63].

Some other CF methods focus on improving the feature representation by directly taking several layers of a pre-trained deep network like VGG [64, 65]. On top of pre-trained convolutional layers, convolution operator tracker (COT) [49] was proposed to integrate multi-resolution convolutional features in different layers. The CREST [66] framework reformulated the CF into a convolutional layer. In addition, Qiang *et al.*

presents an end-to-end lightweight network architecture [67] to learn better features that fit the CF model using offline training. In their work, a CF is treated as a special layer added to a Siamese network. Feature extractor consisting of two convolutional layers, is trained for the online tracking task. We exploit the feature extractor from [67] and further investigate the CF model update problem using the latest reinforcement learning algorithms [68], [69].

The CF-based trackers have demonstrated strong capability in building accurate models with slight online model updating. Also, many proposed new tracking algorithms [66, 70] benefit from the advantage of CF. So we then introduce the correlations filter model in detail.

Definition of the Correlation Filter Model

For CF trackers, a nonlinear regression function $\psi(x) = w^T f(x)$ is trained by minimizing the cost function over training simples x_i and their regressions labels y_i :

$$\arg_w \min \sum_i |\psi(x_i) - y_i|^2 + \lambda \|w\|^2, \tag{1.6}$$

where w is the CF parameters, ψ is a feature extractor, x is a cropped image centered on the target which is used to train the classifier, with exploiting circular matrices structure to learn all the possible shifts of the target. $y \in R^{H \times W}$ is the desired Gaussian shaped response map label and λ is a regularization term.

w can be efficiently solved by transforming (1.6) into the Fourier domain. The Fourier domain representation of f can be calculated as (1.7).

$$W = \frac{\bar{Y} \odot \bar{X}}{\bar{X} \odot X + \lambda}, \tag{1.7}$$

where Y is the Fourier transformation from Gaussian shaped label y, X is the Fourier transformation of x, and the bar means complex conjugation. Operator $\odot$ is the element-wise product.

New search image z around the target in the next frame is cropped with 2 to 4 times of the target size. A response map P in the Fourier domain is obtained by (1.8).

$$P = W \odot \bar{Z}, \tag{1.8}$$

where Z is the Fourier transformation of z. At a new tracking frame, once the CF F is ready, the tracking bounding box center locates at the coordinate that has the maximum response value.

Typically, the numerator A and denominator B of the CF in (1.7) are updated separately using a moving average mechanism.

$$A_t = (1-\eta)A_{t-1} + \eta Y \odot \bar{X}_t, \tag{1.9}$$

$$B_t = (1-\eta)B_{t-1} + \eta X_t \odot \bar{X}_t + \lambda, \tag{1.10}$$

Traditional CF trackers update tracking models frame by frame without considering their tracking results. This may cause an inaccurate model update when occlusion or object missing occurs. Designing a criterion to produce a high-confidence update has been exploded by [71]. Average peak-to-correlation energy (APCE) is proposed to select high-confidence response maps that effectively prevent the CF model from corruption.

Scale estimation is also a critical problem in CF-based tracking, and the translation also plays a vital role in CF methodologies. The convolution operation slides over the search image, as long as the tracking target moves parallel to the plane, the CF model could handle the movements well with convolution operation which is diagonalized in the Fourier domain. Failure in estimating the target scale often leads to severe model drift. One straightforward approach is to apply the learned translation filter at different image scales. That is, the image is first resized by different scale factors, followed by feature extraction. The feature map at each scale is then convolved with the learned filter to compute the response maps. The change in target location and scale can then be estimated by finding the maximum score across all maps.

In our proposed method, instead of calculating an APCE score to decide whether to update the model or not, we introduce a learning algorithm to perform multiple model selection, including the scale estimation.

1.4.4 Dataset and Evaluation metrics

Evaluating the performance of tracking algorithms is difficult because many factors can affect the tracking performance. For better evaluation and analysis of the strengthand weakness of tracking approaches, we use Object Tracking Benchmark [3] to evaluate our proposed tracking approaches. The visual tracking methods are evaluated by two fundamental evaluation categories of performance measures and performance plots. These metrics are briefly described as follows.

- **Center location error (CLE)**: The CLE is defined as the average Euclidean distance between the precise groundtruth locations of target and estimated locations by the visual tracking methods. The CLE is the oldest metric that not only is sensitive to dataset annotation and does not consider tracking failures but also ignores the targets BB and results in significant errors.

- **Accuracy**: To achieve visual tracking accuracy, firstly, the overlap score is calculated as $S = \frac{|b_t \cap b_g|}{|b_t \cup b_g|}$ which b_g, b_t, $\cap$, $\cup$ and $|\cdot|$ represent the ground-truth BB, an estimated BB by a visual tracking method, intersection operator, union operator, and the number of pixels in the resulted region, respectively. By considering a certain threshold, the overlap score indicates the success of a visual tracker in one frame. Then, the accuracy is calculated by the average overlap scores (AOS) during the tracking when a visual tracker's estimations have overlap with the ground-truth ones. This metric jointly considers both location and region to measure the drift rate of the estimated target up to its failure.

- **Area under curve (AUC)**: The AUC score has defined the average success rates (normalized between 0 and 1) according to the pre-defined thresholds. To rank the visual tracking methods based on their overall performance, the AUC score summarizes the AOS of visual tracking methods across a sequence.

  To figure out the performance of visual tracking methods, different methods are usually evaluated in terms of different thresholds to provide more intuitive quantitative comparisons. In the following, these metrics are summarized.

- **Precision plot**: Given the CLEs per different thresholds, the precision plot shows the percentage of video frames in which the estimated locations have at most the specific threshold with the ground-truth locations.

- **Success plot**: Given the calculated various accuracies per thresholds, success plot measures the percentage of frames in which the estimated overlaps and the ground-truth ones have larger overlap than a certain threshold.

For better evaluation and analysis of the strengthand weakness of tracking approaches, the sequences are categorizeby annotating them with the 11 attributes:

- **IV** Illumination Variation - the illumination in the target region issignificantly changed.

- **SV** Scale Variation - the ratio of the bounding boxes of the firstframe and the current frame is out of the range $[1/t_s, t_s], t_s > 1(ts$=2).

- **OCC** Occlusion - the target is partially or fully occluded.

- **DEF** Deformation - non-rigid object deformation.

- **MB** Motion Blur - the target region is blurred due to the motion oftarget or camera.

- **FM** Fast Motion - the motion of the ground truth is larger thantmpixels ($tm = 20$).

- **IPR** In-Plane Rotation - the target rotates in the image plane.
- **OPR** Out-of-Plane Rotation the target rotates out of the imageplane.
- **OV** Out-of-View - some portion of the target leaves the view.
- **BC** Background Clutters - the background near the target has the similar color or texture as the target.
- **LR** Low Resolution - the number of pixels inside the ground-truthbounding box is less than tr ($tr = 400$).

1.5 Deep reinforcement Learning

In recent years, with the combination of deep learning and reinforcement learning (RL), the deep reinforcement learning provides new ideas for computer vision algorithms. Compared with the supervised learning algorithms, RL algorithm has the ability of learning from unlabeled data, and the abilities of interacting with environment and making decisions. In this thesis, we propose a novel model selection approach with RL algorithm to handle the challenging update problem in visual tracking. The basic concepts of reinforcement learning are introduced in this section.

Deep reinforcement learning algorithms have already been applied to various problems arising from different domains. Control policies for robots can be learned by RL directly from real camera outputs [72, 73]. Deep learning enables RL to scale to decision-making problems. The standout success of AlphaGo, which defeated a human world champion in Go, has shown that deep RL can handle complex states and action spaces very well. Also deep RL is applied for many computer vision tasks like objection localization [74], [75], object detection [76], action recognition [77] and person re-identification [78].

Reinforcement learning is the branch of machine learning inspired by behaviorist psychology that is concerned with taking actions and making sequences of decisions to maximize some notion of cumulative reward. As shown in Fig. 1.10.

Reinforcement learning is mainly composed of **Agent**, **Environment**, **State**, **Action**, and **Reward**. When an agent situated in an environment, the agent performs an action, the environment will transition to a new state, and return a reward signal (positive or negative reward) for this new state. Subsequently, the agent performs new actions according to a certain strategy with the environment's feedback. The agent and the environment interact through states, actions, and rewards. An RL algorithm seeks to maximize the agent's total reward, given a previously unknown environment, through a trial-and-error learning process.

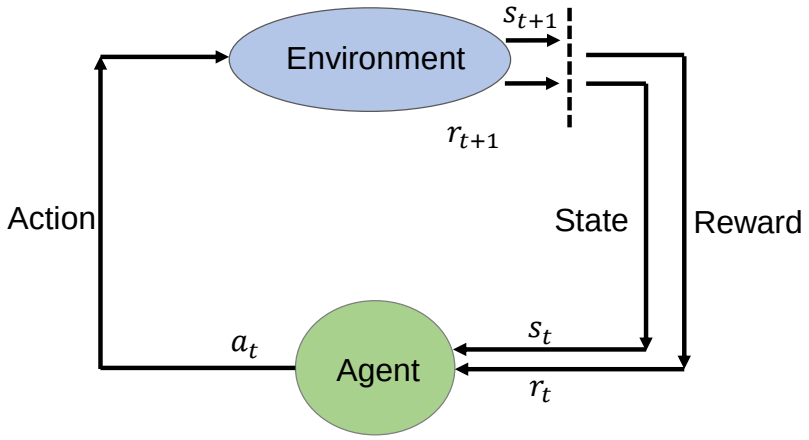


Fig. 1.10: The illustrate figure of reinforcement learning.

The policy gradient is a popular approach in deep reinforcement learning approaches. Policy gradient methods directly learn the policy by optimizing the deep policy networks by concerning the expected future rewards using gradient descent.

Many years ago, Williams et al. [79] proposed an algorithm that was simply using the immediate reward to estimate the value of the policy. Silver et al. [80] proposed a deterministic algorithm to improve the performance and effectiveness of the policy gradient in high-dimensional action space.

1.5.1 Settings of Policy Gradient

The policy gradient methods target modeling and optimizing the policy directly. The policy is usually modeled with a parameterized function respect to θ, $\pi_\theta(a|s)$. The reward function depends on the policy, and different algorithms can be applied to optimize θ for the best reward.

A Markov Decision Process(MDP) $M = (\mathcal{S}, \mathcal{A}, P, r, \gamma, \rho)$ is specified by: a finite state space $\mathcal{S}$; a finite action space $\mathcal{A}$; a transition model P where $P(s'|s, a)$ is the probability of transitioning into state s' upon taking action a in state s; a reward function r where $r(s, a)$ is the immediate reward associated with taking action a in state s; a discount factor $\gamma \in [0, 1)$;

A deterministic policy $\pi : \mathcal{S} \to \mathcal{A}$ specifies a decision-making strategy where the agent chooses actions adaptively based on the current state, i.e., $a_t = \pi(s_t)$. The agent may also choose actions according to a stochastic policy $\pi : \mathcal{S} \to \Delta(\mathcal{A})$, where $\Delta(\mathcal{A})$ is the probability simplex over $\mathcal{A}$. We denote $a_t \sim \pi(\cdot|s_t)$. A policy induces a distribution over trajectories $\tau = (s_t, a_t, r_t)_{t^\infty}^{\infty}$, where s_0 is drawn from the starting state, and for all subsequent timesteps $t, a_t \sim \pi(\cdot|s_t)$ and $s_{t+1} \sim P(\cdot|s_t, a_t)$. The value function $V^\pi : \mathcal{S} \to \mathbb{R}$ is defined as the discounted sum of future rewards starting at state s and

executing π, i.e.

$$V^{\pi}(s) := \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t r\left(s_t, a_t\right) | \pi, s_0 = s\right]. \tag{1.11}$$

The action-value function $Q^{\pi} : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ and the advantage function $A^{\pi} : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ are defined as:

$$Q^{\pi}(s,a) = \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t r\left(s_t, a_t\right) | \pi, s_0 = s, a_0 = a\right], \quad A^{\pi}(s,a) := Q^{\pi}(s,a) - V^{\pi}(s) \tag{1.12}$$

The goal of the agent is to find a policy π that maximizes the expected value from the initial state. Using gradient ascent, we can move the parameters toward the direction suggested by the gradient to find the best θ for that produces the highest return.

$$\nabla_{\theta} V^{\pi_{\theta}}\left(s_0\right) = \mathbb{E}_{s \sim d_{s_0}^{\pi_{\theta}}} \mathbb{E}_{a \sim \pi_{\theta}(\cdot|s)}\left[\nabla_{\theta} \log \pi_{\theta}(a|s) A^{\pi_{\theta}}(s,a)\right] \tag{1.13}$$

where $d_{s0}^{\pi}(s) = \lim_{t \to \infty} P\left(s_t = s | s_0, \pi_{\theta}\right)$ is the probability that $s_t = s$ when starting from s_0 and following policy π_{θ} for t steps.

High variance in gradients makes it difficult to train a deep RL network. Actor-critic methods [80, 81] utilize learned value function as feedback term to guide the training. Trust region policy optimization (TRPO) [68] has been shown to be relatively robust and applicable to domains with high-dimensional inputs. To achieve this, TRPO optimizes a surrogate objective function. Specifically, it optimizes an importance sampled advantage estimate, constrained with a quadratic approximation of KL divergence. And the objective function becomes:

$$V^{\pi_{\theta}}\left(s_0\right) = \mathbb{E}_{s \sim d_{s_0}^{\pi_{\theta_{old}}}} \mathbb{E}_{a \sim \pi_{\theta_{old}}(\cdot|s)}\left[\nabla_{\theta} \frac{\pi_{\theta}(a|s)}{\pi_{\theta_{old}(a|s)}} A^{\pi_{\theta_{old}}}(s,a)\right] \tag{1.14}$$

The latest proximal policy optimization (PPO) [69] algorithm performs unconstrained optimization, requiring only first-order gradient information. Without limitations on the distance between θ_{old} and θ, maximizing $V^{\text{TRPO}}(\theta)$ would lead to instability with extremely large parameter updates and large ratios. PPO imposes the constraint by forcing $r(\theta)$ to stay within a small interval $[1-\varepsilon, 1+\varepsilon]$, where ε is a hyperparameter. We will detail this algorithm in Chapter 4. The objective function of PPO takes the minimum one between the original value and the clipped version and therefore decreases the policy update under extremes, and it results in better rewards. Due to its good performance, PPO is gaining popularity for a range of deep RL tasks. Our work also uses the PPO algorithm to learn a policy for selecting an appropriate CF model for visual tracking.

Employing the deep RL algorithms into computer vision problems could benefit from the experience. In fact, RL has been studied for visual tracking in several recent works [82,

83], [84], [85]. Huang *et al.* succeed in utilizing Q-learning [86] for shallow-level or high-level feature selection [82]. Silver et al. [87] showed that pre-training the policy networks with supervised learning before employing policy gradient could improve the performance. Inspired by their observation, Yun *et al.* then proposed an action-decision network[83] used policy gradient learning and trained action dynamics for tracking with annotated visual tracking sequences. Luo *et al.* proposed an active tracking scheme trained in simulators by reinforcement learning [88]. Our work is distinct from these existing works in that we are studying the model update issue in visual object tracking with reinforcement learning.

1.6 Overview of This Thesis

1.6.1 Major Contributions

The major contributions of the research reported in this thesis are summarized as follows:

- We propose a novel Individual Aggregation Network (IAN) that can not only accurately localize pedestrians but also minimize feature representations of intra-person variations. In particular, IAN is built upon the state-of-the-art object detection framework, i.e., Faster R-CNN [89], so that high-quality region proposals for pedestrians can be produced in an online manner for person search. In addition, to relieve the negative effect caused by various visual appearances of the same individual, a novel center loss [32] that can increase the intra-class compactness of feature representations is introduced. The center loss encourages learned pedestrian representations from the same class to share similar feature characteristics. The IAN model can be embedded in any CNN-based person search framework for improving performance.

- We propose an ensemble siamese tracker, where the final similarity score is also affected by the similarity with tracking results in recent frames instead of solely considering the first frame. Tracking results in 25 recent frames are used to adjust the model for a continuous target change. Meanwhile, we combine adaptive candidate sampling strategy and large displacement optical flow method with our method to improve the performance further.

- We propose a novel approach for selecting an optimal model among multiple CF models that are updated and maintained in parallel. This approach addresses a number of concerns that arise from a single CF model, such as drift; We propose a reinforcement learning-based approach for optimal model selection. To the best of

our knowledge, this is the first time that reinforcement learning is utilized for model selection among multiple CF models; We utilize a lightweight feature extractor and proposed a small decision network so that the proposed approach can be deployed in realtime applications, where the frame rates are high;

1.6.2 Brief Summary of the Remaining Chapters

In this chapter, the final summary of this thesis will be presented, followed by the future work for the research in relevant domains.

Chapter 2: In this chapter, we propose a novel Individual Aggregation Network (IAN) that can accurately localize persons by learning to minimize intra-person feature variations. Person search in real-world scenarios is a new challenging computer version task with many meaningful applications. IAN is built upon the state-of-the-art object detection framework, i.e., faster R-CNN, so that high-quality region proposals for pedestrians can be produced in an online manner. In addition, to relieve the negative effect caused by varying visual appearances of the same individual, IAN introduces a novel center loss that can increase the intra-class compactness of feature representations. The engaged center loss encourages persons with the same identity to have similar feature characteristics. Extensive experimental results on two benchmarks, i.e., CUHK-SYSU and PRW, well demonstrate the superiority of the proposed model.

Chapter 3: In this chapter, we propose a siamese ensemble tracker that extends a Siamese INstance search Tracker (SINT) with a model updating mechanism to improve its tracking robustness. SINT uses convolutional neural network (CNN) features and compares the new frame features with the target features in the first frame. The candidate region with the highest similarity score is considered as the tracking result. However, SINT is not robust against large target variation because the matching model is not updated during the whole tracking process. To combat this defect, we propose an Ensemble Siamese Tracker (EST), where the final similarity score is also affected by the similarity with tracking results in recent frames instead of solely considering the first frame. Tracking results in recent frames are used to adjust the model for a continuous target change. Meanwhile, we combine adaptive candidate sampling strategy and large displacement optical flow method with EST to improve further its performance, which is named EST+. We test the proposed EST and EST+ on a standard tracking benchmark OTB. It turns out the average overlap ratio of EST and EST+ increase 3.55% and 2.26% respectively compared with SINT on OTB 100. The time complexity of EST and EST+ is low, which can almost achieve realtime application.

Chapter 4: In this chapter, we propose a novel approach to address the correlation filter update problem. In our approach, we update and maintain multiple correlation filter

models in parallel, and we use deep reinforcement learning for the selection of an optimal correlation filter model among them. To facilitate the decision process in an efficient manner, we propose a decision-net to deal with target appearance modeling, which is trained through hundreds of challenging videos using proximal policy optimization and a lightweight learning network. An exhaustive evaluation of the proposed approach on the OTB100 and OTB2013 benchmarks show that the approach is effective enough to achieve the average success rate of 62.3% and the average precision score of 81.2%, both exceeding the performance of traditional correlation filter based trackers.

For each of these chapters mentioned above, we have tried to make them self-contained. Therefore, some of the crucial contents, demonstrations, model definitions, and depictive illustrations might be reiterated in the following chapters when necessary.

Chapter 2

Feature learning in the image-based person re-identification

This chapter starts by introducing the motivation behind this work and related works in person re-identification. Then we detailed the proposed method with the designing of the network structure and loss function. A discussion on the usage of dropout is presented, and a performance gain can be obtained by removing the dropout. Afterward, the implementation detail is presented, specifying the training and testing phases. Finally, experimental results on two large scale datasets are provided.

2.1 Motivation

Technically, a person re-identification system for video surveillance consists of three components, including person detection, person tracking, and person retrieval. While the first two components are independent computer vision tasks, most person re-identification studies focus on the third component. Numerous re-identification algorithms, as well as datasets, have been proposed during the past decades, and the performance on these benchmarks has been improved substantially. All these algorithms focus on the third component of the pipeline, assuming the person/pedestrian detection is already available. In other words, a query person is matched with cropped pedestrians in the gallery instead of searching for the target person from whole images. In reality, perfect pedestrian bounding boxes are unavailable in surveillance scenarios. In addition, existing pedestrian detectors unavoidably produce false alarms, misdetections, and misalignments. All these factors compromise the re-identification performance. Therefore, current re-identification algorithms cannot be directly applied to real surveillance systems, where we need to search for a person from whole images, as shown in Fig. 2.1.

While the majority of person re-identification works engage boxes manually annotated or produced by a fixed detector in their applications, it is necessary to study the impact

of pedestrian detectors on re-identification accuracy. Specifically, considering detection and re-identification jointly leads to higher person search accuracy than optimizing them separately. To the best of our knowledge, end-to-end deep learning for person search [1] (E2E-PS) is the first work to introduce an end-to-end deep learning framework to jointly handle the challenges from both detection and re-identification. Thereby, the detector and re-identification parts can interact with each other so as to reduce the influence of detection misalignments.

In E2E-PS, the re-identification feature learning exploits a modified softmax loss. Early works show that such kind of identification task could greatly benefit the feature learning [14]. Meanwhile, it is found that the identification task increases the inter-personal variations by drawing features extracted from different identities apart, while the verification task reduces the intra-personal variations by pulling features extracted from the same identity together [15]. Inspired by this, softmax loss and contrastive loss are jointly used for feature learning, leading to better performance than the sole softmax loss. But we can not directly introduce such verification tasks into the person search faster R-CNN framework used in E2E-PS. Because the pedestrians appearing in each image are random, sparse, and unbalanced, which make it difficult to group the positive and negative pairs for training with verification loss such as contrastive loss within the Faster R-CNN framework.

In this work, to address this critical issue, we propose a novel Individual Aggregation Network (IAN) that can not only accurately localize pedestrians but also minimize feature representations of intra-person variations. In particular, we built IAN upon the state-of-the-art object detection framework, i.e., Faster R-CNN, so that high-quality region proposals for pedestrians can be produced in an online manner for person search. In addition, to relieve the negative effect caused by various visual appearances of the same individual, a novel center loss that can increase the intra-class compactness of feature representations is introduced. The center loss encourages learned pedestrian representations from the same class to share similar feature characteristics. The IAN model can be embedded in any CNN-based person search framework for improving performance.

In particular, the center loss is able to increase intra-class feature compactness without requiring to aggregate positive and negative verification samples. Center loss tracks the feature centers of all classes, and these feature centers are constantly updated based on the recently observed class samples. Meanwhile, it manages to pull the sample features towards each class center that this sample belongs to. This process is illustrated in Fig. 2.2 During the model development, we found that neural networks with dropout are not compatible with center loss.

In this chapter, we study this phenomenon in both analytic and experimental ways.

Fig. 2.1: Person search from whole images without cropping out persons. The left column is probe/query image, other columns are gallery images without cropped pedestrians. The green bounding boxes are searching results. To find the right person in the gallery images, we need to detect all the persons within the image, and compare the detected persons with the probe image.

We believe this finding could be useful guidance for neural network framework design in the community, which is one of our contributions.

2.2 Proposed method

In practical person search applications, pedestrian bounding boxes are unavailable, and the target person needs to be found from the whole images. Targeting this problem, IAN is built upon the state-of-the-art object detection framework, i.e., faster R-CNN so that reasonable region proposals for pedestrians can be produced in an online manner for person search. It should be noted that faster R-CNN could be built on top of any backbone networks, such as AlexNet, VGGNet, GoogLeNet, and ResNet. The backbone network is divided into 2 parts, layers before RPN form the first part network, while layers after RPN are the second part. The proposed IAN framework is shown in Fig. 2.1, and it is elaborated as follows.

1. In the training phase, arbitrary size images with ground truth pedestrian bounding boxes and identifications are input into the first part backbone network, i.e., ResNet

2. The region proposal network (RPN), is built on top of the feature maps gener-

ated with the first part, network to predict pedestrian bounding boxes. The RPN is trained with ground truth pedestrian bounding boxes, using two loss layers, i.e., anchor classification and anchor regression. Besides the candidate boxes generated by the region proposal network (RPN boxes), the ground truth (GT) pedestrian bounding boxes are also used together at the network training stage. At the test stage, only RPN boxes are available.

3. All the candidate boxes (RPN+GT boxes at the training stage, RPN boxes at test stage) are used for ROI pooling to generate a feature vector for each candidate box. These features are again convolved with the second part backbone network.

4. Two sibling fully connected layers are utilized separately, one to produce the final feature vector *feat* to compute feature distance, and the other to produce bounding box locations. At the training stage, feature vectors of all candidates boxes (RPN+GT boxes) are fed into the softmax loss layer, while only feature vectors of ground truth pedestrian boxes (GT boxes) are fed into the center loss layer. The softmax variant random sampling softmax (RSS) is used for training.

Overall, compared with the previous person search method E2E-PS, the proposed IAN generates more discriminative feature representations. In IAN, using softmax loss together with center loss within the faster R-CNN framework leads to better feature representations than solely using softmax loss in Meanwhile, the VGGNet used in E2E-PS contains dropout layers which are intrinsically not compatible with the center loss. In our IAN, we use the state-of-the-art residual network In addition to solving the compatibility issue with center loss, replacing VGGNet with the residual network also offers better discrimination power with a lower computational cost.

2.2.1 Random sampling softmax loss and Center loss

In this section, we introduce the loss function design of our proposed method. The Random sampling softmax loss is used to better distinguish between identities and false alarms predicted by the detector net. And the center loss encourages intra-class variation compactness. We combine these two losses in this work to compact intra-class variations and separable inter-class differences.

Random sampling softmax Loss (RSS)

For the original softmax loss, the gradients could favor only a few numbers of classes that appear in a minibatch, while severely suppress the other classes. The RSS loss layer

solves this problem by randomly selecting a subset of softmax neurons for each input sample to compute the loss and gradients. The detailed formulation is given below.

Suppose the target classes are from 1 to $C+1$, where class $C+1$ is the background, and the others are the identities. Denote each data sample by $\{x, t\}$, where $x \in R^{C+1}$ is the classifier scores (input of the softmax) and t is a binary vector which representing the label. Then the original softmax loss can be written as:

$$l = -\sum_{i=1}^{C+1} t_i \log y_i, \qquad \text{where } y_i = \frac{e^{x_i}}{\sum_{j=1}^{C+1} e^{x_j}} \tag{2.1}$$

The RSS loss will randomly select $K (K \ll C+1)$ dimensions from x and t to compute the loss and gradients. Suppose the selected indices are $i_1, i_2, \ldots, i_K$, the sampled classifier scores and label vector can be denoted by $\tilde{x} = (x_{i_1}, x_{i_2}, \ldots, x_{i_K})^T$ and $\tilde{t} = (t_{i_1}, t_{i_2}, \ldots, t_{i_K})^T$ Then the RSS loss function is defined as:

$$\tilde{l} = -\sum_{i=1}^{K} \tilde{t}_i \log \tilde{y}_i, \qquad \text{where } \tilde{y}_i = \frac{e^{\tilde{x}_i}}{\sum_{j=1}^{K} e^{\tilde{x}_j}} \tag{2.2}$$

This helps the net to better distinguish between identities and false alarms predicted by the detector net.

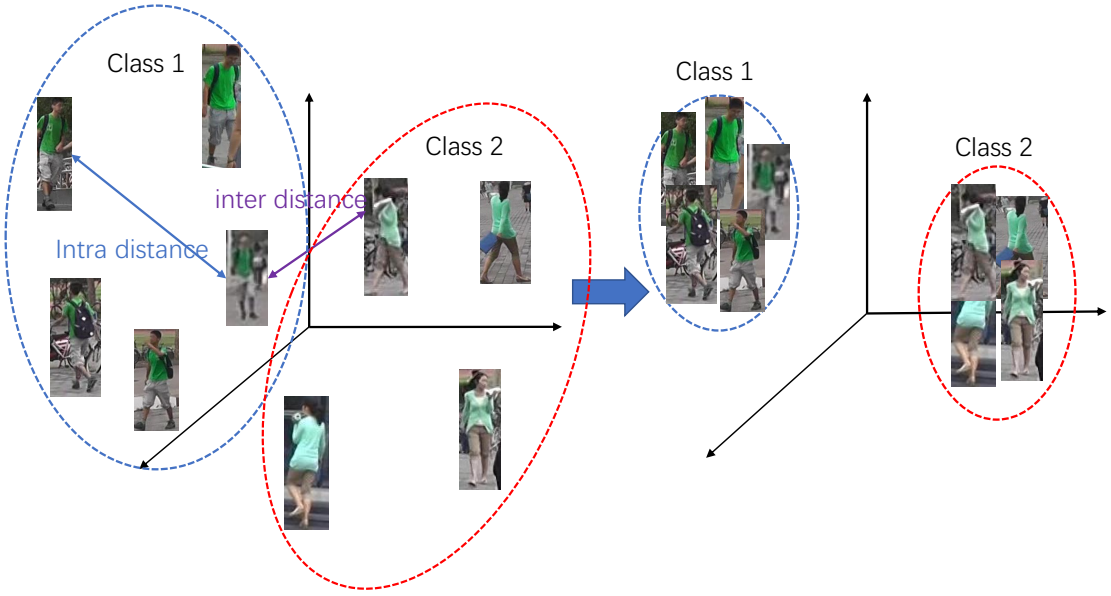


Fig. 2.2: The objective of center loss is to reduce the intra-class distance by pulling the sample features towards each class center. Left side: feature distance without center loss; right side: feature distance using center loss.

Center Loss

Both compact intra-class variations and separable inter-class differences are essential for discriminative features. However, the softmax loss only encourages the separability of features. Contrastive loss [90, 15] and triplet loss [9], that respectively construct loss functions for image pairs and triplets, are possible solutions to encourage intra-class variation compactness. For contrastive loss, an equivalent amount of positive and negative image pairs are required, whereas for triplet loss, two images among each triplet should belong to the same class/identification with one belonging to different class/identification. However, for the faster R-CNN based person search framework, it is a non-trivial task to form such image pairs and triplets within the input mini-batch. The pedestrians within each image belong to different identifications. Meanwhile, the pedestrians appearing in each image are random, sparse, and unbalanced. Within the mini-batch of faster R-CNN, it is difficult to form a balanced number of positive pedestrian pairs as negative pairs.

On the other hand, employing center loss [32] is able to avoid the need for aggregating positive and negative pairs. In the proposed IAN network, the center loss is applied together with softmax loss to generate feature representations. The center loss function is defined as follows.

$$\mathcal{L}_c = \frac{1}{2}\sum_{i=1}^{m} \|x_i - c_{y_i}\|^2 \tag{2.3}$$

where $x_i \in \mathbb{R}^d$ is the feature vector of pedestrian box i, which belongs to class y_i, and $c_{y_i} \in \mathbb{R}^d$ denotes the y_i-th class center of features. The softmax loss forces the features of different classes to stay apart. The center loss pulls the features of the same class closer to their centers. Hence the feature discriminative power is highly enhanced. With the center loss, the overall network loss function is defined as:

$$\mathcal{L} = \mathcal{L}_4 + \lambda\mathcal{L}_c \tag{2.4}$$

where $\mathcal{L}_4$ is the summation of 4 loss functions in faster R-CNN, which includes the softmax loss for perosn identification classification, and λ is the weight of the center loss.

Ideally, c_{y_i} should be constantly updated as the network parameters are being updated. In other words, we need to take the entire training set into account and average the features of every class in each iteration, which is inefficient and impractical. In fact, we learn the feature center of each class one by one. In the training process, we simultaneously update the center and minimize the distances between the features and their corresponding class centers.

The center c_{y_i} is updated based on each mini-batch. In each iteration, the centers are computed by averaging the features of the corresponding classes. Meanwhile, to avoid large perturbations caused by a few mislabelled samples, we use a scalar $\alpha \in [0, 1]$ to

control the learning rate of the centers. The gradients of $\mathcal{L}_c$ with respect to x_i and the updating equation of c_{y_i} are computed as:

$$\frac{\partial \mathcal{L}_c}{\partial x_i} = x_i - c_{y_i} \tag{2.5}$$

$$\Delta c_j = \frac{\sum_{i=1}^{m} \delta(y_i = j) \cdot (c_j - x_i)}{1 + \sum_{i=1}^{m} \delta(y_i = j)} \tag{2.6}$$

where $\delta(\textit{condition}) = 1$ if the *condition* is satisfied, and otherwise $\delta(\textit{condition}) = 0$.

2.2.2 Dropout

In our study, we notice that neural networks with dropout are not compatible with the center loss. For example, when the proposed IAN is deployed on VGGNet with 3 dropout layers, its person search mAP performance on the CUHK-SYSU person search dataset [1] is about 10% lower than the results obtained by removing all the dropout layers.

Dropout is a technique for addressing overfitting problems [91]. The key idea of dropout is to randomly drop units, along with their connections, from the neural network during training. Since the dropout randomly drops units, it creates uncertainty for the features. In other words, when image features are extracted using the same network with dropout, the obtained features for the same image might be quite different in the different network forward computation instances. This is contradicting with center loss, which punishes intra-class variations.

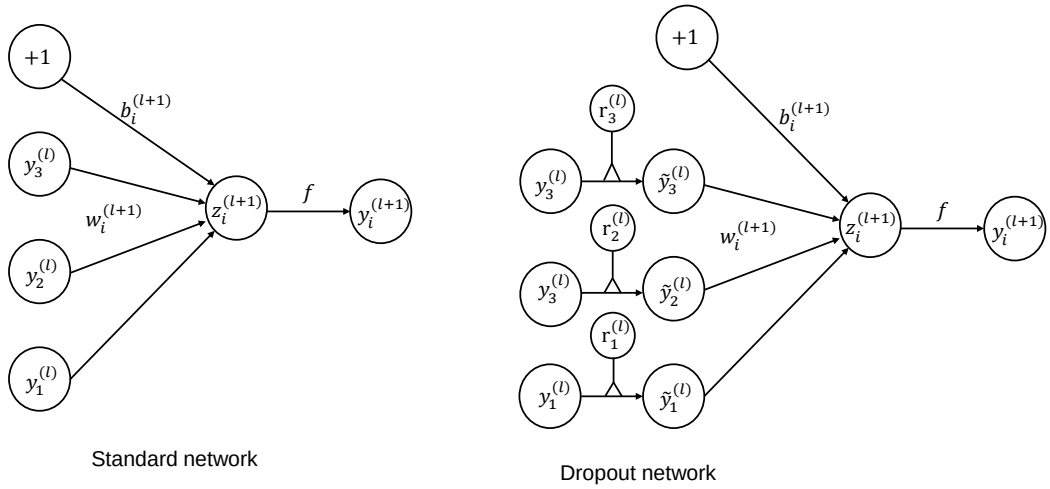


Fig. 2.3: Standard network and Dropout network.

The dropout is usually deployed after the fully connected layer, as in VGGNet. Let $z^{(l)}$ denote the vector of inputs into layer l, and $y^{(l)}$ denote the vector of outputs from layer l. $W^{(l)}$ and $b^{(l)}$ are the weights and biases at layer l, respectively. The feed-forward operation of a standard neural network can be described as

$$z_i^{(l+1)} = w_i^{(l+1)} y^{(l)} + b_i^{(l+1)} \tag{2.7}$$

$$y_i^{(l+1)} = f(z_i^{(l+1)}) \tag{2.8}$$

where f is any activation function, for example sigmoid or ReLu function. With dropout, the feed-forward operation becomes

$$r_j^{(l)} \sim \textit{Bernoulli}(p) \tag{2.9}$$

$$\tilde{y}^{(l)} = r^{(l)} * y^{(l)} \tag{2.10}$$

$$z_i^{(l+1)} = w_i^{(l+1)} \tilde{y}^{(l)} + b_i^{(l+1)} \tag{2.11}$$

$$y_i^{(l+1)} = f(z_i^{(l+1)}) \tag{2.12}$$

Here $*$ denotes an element-wise product. For any layer l, $r^{(l)}$ is a vector of independent Bernoulli random variables each of which has probability p of being 1.

To illustrate that dropout is not compatible with the center loss, let us take one example. Assume input image samples I_i and I_j are the same and belong to the same pedestrian/class. We assume layer $l+1$ is a fully connected layer with dropout, the output of layer $l+1$ is input into the center loss layer. Since image samples I_i and I_j are the same, we could have $y^{(l)}(I_i) = y^{(l)}(I_j)$. The target of the center loss is to have similar features for the same class, i.e., $y^{(l+1)}(I_i) = y^{(l+1)}(I_j)$. Considering (2.10)(2.12), it is equivalent as $r^{(l)}(I_i) * y^{(l)}(I_i) = r^{(l)}(I_j) * y^{(l)}(I_j)$. Here $r^{(l)}(I_i)$ and $r^{(l)}(I_j)$ are vectors of independent *Bernoulli* random variables, leading to $r^{(l)}(I_i) \neq r^{(l)}(I_j)$. Therefore, to have $y^{(l+1)}(I_i) = y^{(l+1)}(I_j)$, the only solution is $y^{(l)}(I_i) = y^{(l)}(I_j) = \vec{0}$. However, zero feature cannot properly represent the image samples. From the above simple example, we could conclude that dropout is not compatible with center loss, which is consistent with our experimental verification.

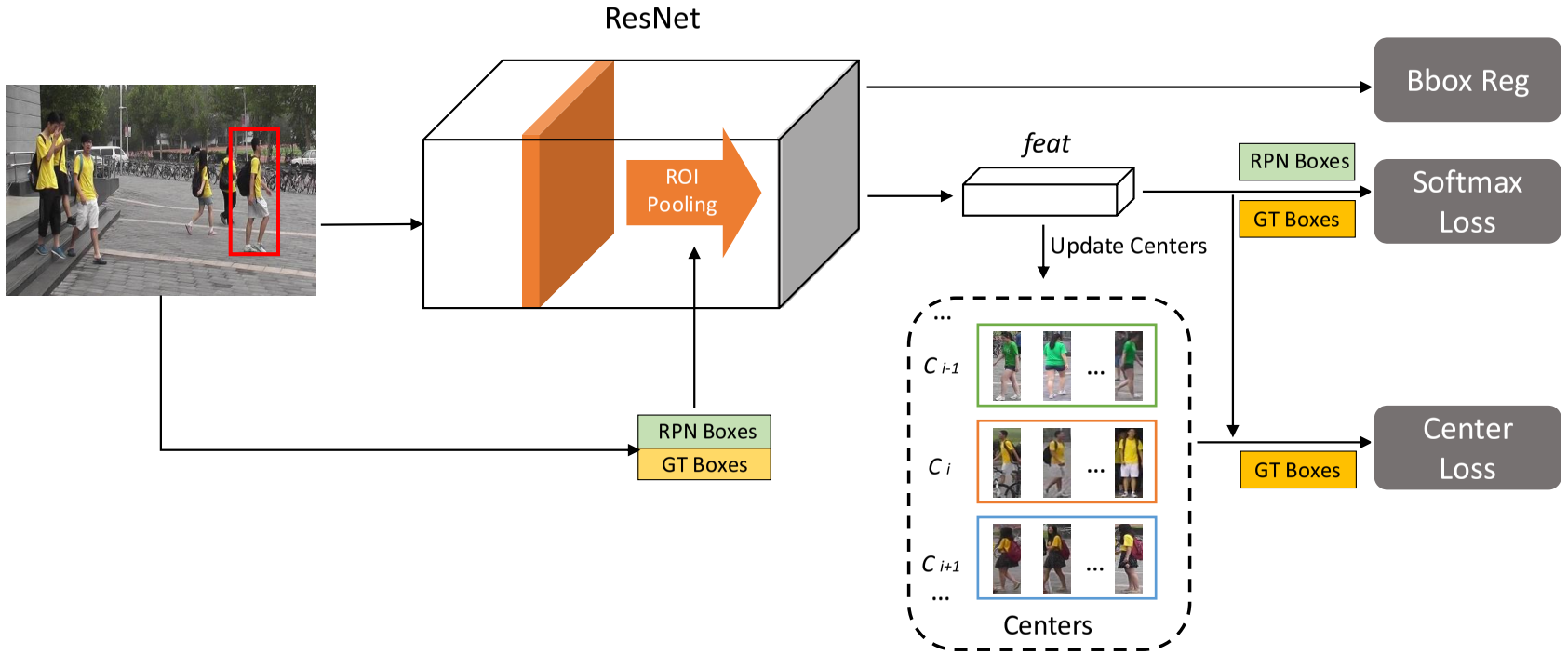


Fig. 2.4: Overview of our IAN network training framework. Images containing pedestrians are input into the network. The first part of the backbone network is used to get image features. Pedestrian bounding boxes generated by region proposal network (RPN boxes), together with the ground truth pedestrian bounding boxes (GT boxes), are used for ROI pooling to generate a feature vector for each box. The second part of the backbone network uses ROI pooling features as input. Two fully connected layers are utilized separately, one to produce the final feature vector *feat* to compute feature distance, and another to produce bounding box locations. Feature vectors of all candidate boxes (RPN + GT boxes) go into the random sampling softmax loss layer, while only feature vectors of ground truth pedestrian boxes (GT boxes) go into the center loss layer.

2.3 Implementation Details

2.3.1 Training Phase

During the network training phase, the network is trained to detect pedestrians and produce discriminative features for re-identification. In our network, 5 loss functions are used. The smoothed-L1 loss [92] is used for the two bounding box regression layers. A softmax loss is used for the pedestrian proposal module, which classifies pedestrian and non-pedestrian. For the re-identification feature extraction part, we deploy both random sampling softmax [1] and center loss [32]. Here it is important to note that only features of ground truth pedestrian boxes are input into the center loss layer. This helps to avoid sample noise. The overall loss is the sum of all five loss functions, and its gradient w.r.t. the network parameters is computed through backpropagation.

To speed up the network convergence process, the training process includes three steps:

1. We crop ground truth bounding boxes for each training person and randomly sample the same number of background boxes. Then we shuffle the boxes, resize them to 224×224, and fine-tune the residual network model (ResNet-101 and ResNet-50) to classify the candidate boxes. The output feature size of ROI-pooling layer in Fig. 2.2.2 is 7×7. To ensure the same feature size, we add one 2×2 pooling layer to the residual network.

2. We fine-tune the model resulting from the above step. Unlike the previous step, the whole images with GT pedestrian bounding boxes and identification annotations are used for the fine-tuning process. Four loss layers excluding the center loss are used in this fine-tuning process.

3. We fine-tune the model obtained in Step 2 with all 5 loss layers, including the center loss. The input images and label annotations are the same as those in Step 2.

2.3.2 Test Phase

The test phase is similar to that in [1]. For each gallery image, we get the features (*feat*) of all the candidate pedestrians by performing the network forward computation once. Whereas for the query image, we replace the pedestrian proposals with the given bounding box and then do the forward computation to get its feature vector (*feat*). Finally, we compute the pairwise Euclidean distances between the query features and those of the gallery candidates. The person similarity level is evaluated based on the Euclidean distances.

2.4 Experiments

In this section, we introduce the evaluation metrics and experiment train/test settings and then report the results on CUHK-SYSU and PRW datasets.

Dataset and Evaluation Metrics We use the benchmark datasets, i.e., both the CUHK-SYSU person search dataset [1] and PRW dataset [13] in our experiment. Both mean Averaged Precision (mAP) and top-1 matching rate metrics are used. A candidate window is considered as positive if it overlaps with the ground truth larger than 0.5, which is the same as the setup in previous works [1, 13].

CUHK-SYSU dataset is a large scale and scene-diversified person search dataset, which contains 18,184 images, 8,432 persons, and 99,809 annotated bounding boxes. Each query person appears in at least two images. Each image may contain more than one query person and many background people. The dataset is partitioned into a training set and a test set. The training set contains 11,206 images and 5,532 query persons. The test set contains 6,978 images and 2,900 query persons. The training and test sets have no overlap on images or query persons. The identifications in CUHK-SYSU dataset is in the range of $[-1, 5532]$, with -1 being unknown persons, and 5,532 being background. Boxes with identification -1 do not go into the random sampling softmax (RSS). Neither -1 nor 5,532 goes into the center loss layer because unknown persons and background are not as unique as other identifications.

In the PRW dataset, a total of 11,816 frames are manually annotated to obtain 43,110 pedestrian bounding boxes, among which 34,304 pedestrians are annotated with an identifications ranging from 1 to 932, and the rest are assigned an identification of -2. The PRW dataset is divided into a training set with 5,704 frames and 482 identifications and a test set with 6,112 frames and 450 identifications. Similar to that in CUHK-SYSU dataset, unknown persons, and background does not go into the center loss layer. Boxes with identification -2 do not go into the random sampling softmax (RSS).

Our ablation study is based on the CUHK-SYSU dataset, so as to provide more comprehensive performance comparisons with state-of-the-art methods, such as E2E-PS [1] and JDI-PS [2].

Training / Testing Settings We build our framework on two residual networks, i.e., ResNet-101 and ResNet-50 [93]. For ResNet-101, the pedestrian proposal the network is connected after layer res4b22, while for ResNet-50, it is connected after layer res4f. In the following experiments, the default network is ResNet-101 if not specified. For training Step 1 described in Section 2.3.1, the learning rate is 0.001 with 20k iterations and batch size being 8. For training Step 2, 120k iterations are used. The initial learning rate

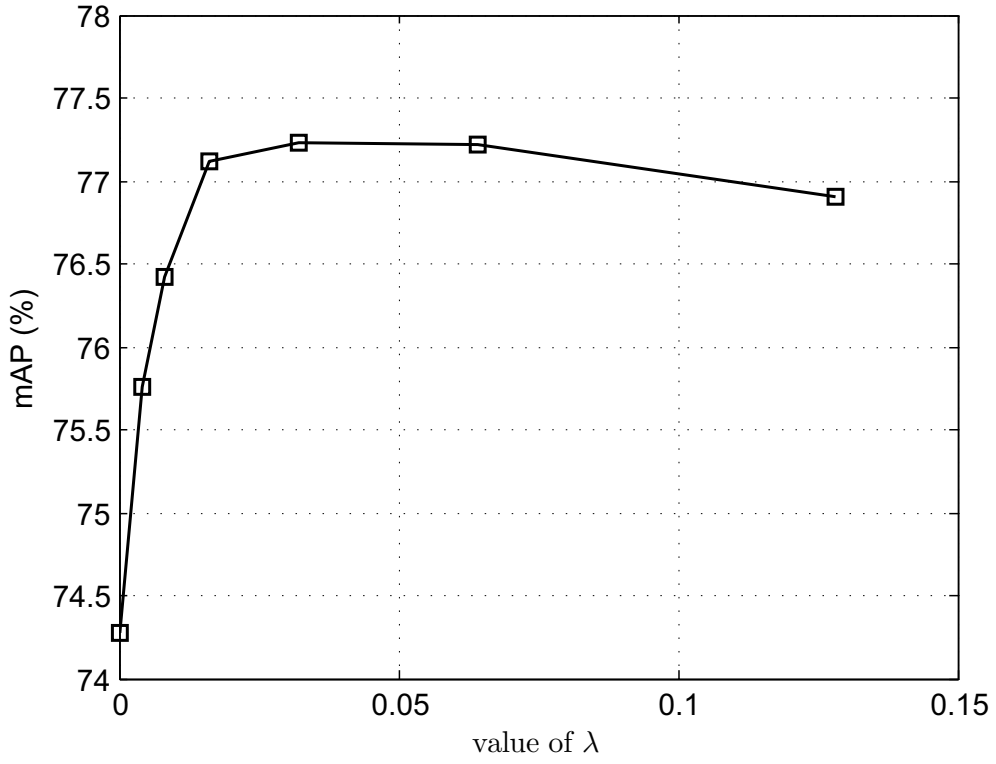


Fig. 2.5: The mAP accuracy of person search on CUHK-SYSU [1] validation set using different center loss weight λ.

is 0.001 and decreased by a factor of 10 after 100k iterations. For training Step 3, the learning rate is 0.0001 with 20k iterations. For both steps 2 and 3, the batch size is 2 due to high memory cost. The networks are trained on NVIDIA GeForce TITAN X GPU with 12GB memory. Our implementation is based on the publicly available Caffe framework [94].

For testing the CUHK-SYSU dataset, in order to evaluate the influence of gallery size, different gallery size is used, including $\{50, 100, 500, 1000, 2000, 4000\}$. In the following experiments, we will report the performance based on the test protocol where the gallery size is 100 if not specified. Each image contains 5.3 background persons on average. If the gallery size is set to 100, a query person has to be distinguished from around 530 background persons and thousands of non-pedestrian bounding boxes, which is challenging. While for testing the PRW dataset, all 6,112 frames in the test set are used as the gallery, which is challenging.

2.4.1 Results on CUHK-SYSU Dataset

Experiment on Parameter λ. The hyperparameter λ controls the weight of the center loss over the whole network loss function. It is essential to our model. So we conduct one experiment to investigate the sensitiveness of the proposed approach with respect to λ. We vary λ from 0 to 0.128 to learn different models. The training dataset is equally divided into 5 equal folds, use 4 of them for training, and 1 for validation. Cross-validation is deployed. The person search accuracies of these models on CUHK-SYSU [1] validation set are shown in Fig. 2.5. It is very clear that it is not a good choice simply without using the center loss (in this case $\lambda = 0$), leading to poor person search mAP performance. Proper choice of the values, e.g., $\lambda = 0.032$, can improve the person search accuracy of the deeply learned features. We also observe that the person search performance of our model remains largely stable across a wide range of $[0.016, 0.128]$. Meanwhile, it is also observed that a similar trend is obtained for the top-1 accuracy. Thus, in the following experiments, we set the λ value as 0.032. It is interesting to note that for the VGGNet without dropout, similar optimal λ is obtained.

Overall Person Search Performance. The results of IAN and benchmarks under two evaluation metrics are summarized in Table 2.1. We compare our performance with end-to-end deep learning for person search (E2E-PS) method [1], and joint detection and identification feature learning for person search (JDI-PS) method [2], because of their superior performance. As reported in [2], JDI-PS method [2] attains much better performance than separating pedestrian detection ([95], [96]) and re-identification (for examples, BoW [97]+ Cosine similarity, LOMO+XQDA [98]).

With ResNet-101, more than 7% gain is obtained compared with [1] for both mAP and top-1 accuracy. To demonstrate the importance of center loss in IAN, we also report the performance of E2E-PS [1] when the VGGNet is replaced with ResNet-101 and ResNet-50. It is observed that about 3% gain for the two metrics is obtained only because of the center loss.

Compared with JDI-PS [2], our gain is not very big. When using the Res-50 backbone network, our accuracy is 80.07%, 1.37% higher than that of JDI-PS [2]. However, compared with E2E-PS [1], our gain is about 3%. In fact, if we remove the center loss in our IAN network, it degrades to the E2E-PS [1] network. Therefore, we should consider E2E-PS [1] as our baseline.

On the other hand, it should be noted that in JDI-PS [2], a specifically designed loss function called Online Instance Matching (OIM) was used. OIM loss also helps to increase the intra-class compactness, which is very important for person search. We think that the main reason that our design leads to better performance than JDI-PS [2] is due to

Table 2.1: Comparisons between IAN with E2E-PS [1] and JDI-PS [2].

Method	E2E-PS [1] (VGGNet)	E2E-PS [1] (ResNet-50)	E2E-PS [1] (ResNet-101)	JDI-PS [2] (ResNet-50)	IAN (ResNet-50)	IAN (ResNet-101)
mAP (%)	69.69	73.13	74.28	75.5	76.28	**77.23**
top-1 (%)	72.97	77.34	78.17	78.7	80.07	**80.45**

Table 2.2: The person search performance if all positive pedestrian boxes are input into the center loss layer (IAN with all boxes).

Method	IAN with all boxes	IAN
mAP (%)	74.70	77.23
top-1 (%)	77.72	80.45

one key reason. In IAN, different candidate boxes are treated differently. Only features of ground truth pedestrian boxes are input into the center loss layer, while features of both ground truth pedestrian boxes and RPN boxes should be input to the classification loss (softmax). While, in JDI-PS [2], there is only one loss function (OIM), features of both the ground truth pedestrian boxes and RPN boxes go to the OIM loss. RPN boxes include a lot of background regions. We believe enforcing intra-class compactness with RPN boxes is harmful to feature learning due to large background variations.

Input of Center Loss. In our proposed method, only features of ground truth pedestrian boxes are input into the center loss layer. This scheme is verified by experimental results. To do this, we input all positive pedestrian boxes (excluding background and unknown persons with id -1) into the center loss layer. Note that positive pedestrian boxes refer to candidate boxes overlapping with ground truth pedestrian boxes higher than the threshold, i.e., 0.5, so they include both ground truth pedestrian boxes and RPN boxes. The obtained results with such a scheme are lower than that uses features of ground truth pedestrian boxes, as reported in Table 2.2. This is because the objective of center loss is to increase intra-class feature compactness, but features of different positive boxes of the same pedestrian are dissimilar as they cover different regions with various background information.

Center Loss with VGGNet. In Section 2.2.2, analysis to avoid dropout is given. We also study this phenomenon with experiments. The VGGNet model provided in [1], where

Table 2.3: Person search performance using VGGNet (dropout) and center loss together.

Iteration	0	$10,000$	$20,000$	$30,000$	$40,000$
mAP (%)	69.69	67.38	64.12	62.55	60.73
top-1 (%)	72.97	71.31	69.03	66.79	66.21

Table 2.4: Comparison between IAN and E2E-PS [1] for VGGNet with all dropout layers removed.

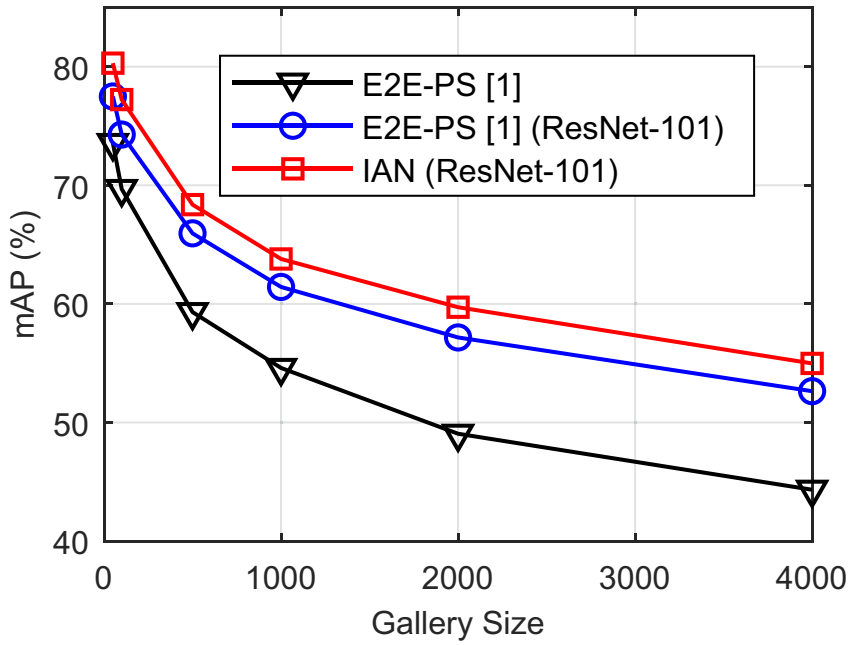
Method	E2E-PS[1] (VGGNet)	E2E-PS[1] (VGGNet no dropout)	IAN (VGGNet)
mAP (%)	69.69	71.21	**73.65**
top-1 (%)	72.97	74.48	**76.14**

dropout layers are used, is fine-turned with the center loss with loss weight 0.0032. The testing results with the fine-turned models are reported in Table 2.3. It is observed that by increasing the iteration number, the performance is decreased constantly. With $40,000$ iterations, almost 9% mAP is dropped compared with models without center loss. The importance of replacing VGGNet with ResNet is demonstrated with this experiment.

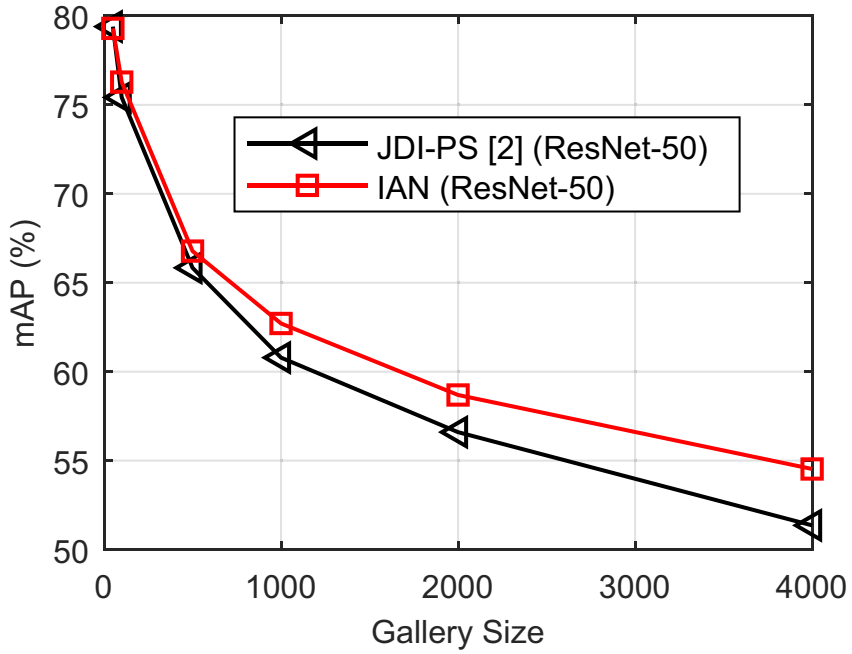
We remove all the dropout layers in VGGNet, and test E2E-PS[1] and our IAN. The obtained results are reported in Table 2.4. It is interesting to see that removing the 3 dropout layers in VGGNet leads to a slightly better person search performance. Our IAN with center loss leads to about 2% performance gain compared with E2E-PS[1] for both mAP and top-1 accuracy if both remove the dropout layers. By comparing the results in Table 2.3 and 2.4, it is evident that dropout and center loss are not compatible. The experimental results support our analysis in Section 2.2.2.

Effects of Gallery Size. The task of person search is more challenging when the gallery size increases. We vary the gallery size from 50 to 4,000, and test our approach, E2E-PS [1] with both VGGNet and ResNet-101, and JDI-PS [2]. The obtained mAPs for various gallery sizes are reported in Fig. 2.6. As expected, the mAP decreases with the increase in gallery size. Meanwhile, for various gallery sizes, our approach outperforms E2E-PS [1] with both VGGNet and ResNet-101 significantly. For large gallery size 4,000, the mAP gain over E2E-PS [1] is more than 10%. Meanwhile, it is also observed from

Fig. 2.6.(b) that IAN outperforms JDI-PS [2] with good gain for various gallery size. For large gallery size, i.e., 4,000, the mAP gain is 3%. It is worth noticing that the comparison is fair because both use the ResNet-50 network.



(a)



(b)

Fig. 2.6: Person search performance comparison for various gallery size. (a) Comparing IAN with E2E-PS [1]; (b) Comparing IAN with JDI-PS [2].

Table 2.5: Experimental results of three solutions on the occlusion subset, low-resolution subset.

Method	E2E-PS [1] VGGNet		E2E-PS [1] (Res-101)		IAN (Res-101)	
	mAP	top-1	mAP	top-1	mAP	top-1
Low-Res	46.11	51.03	47.91	52.07	**52.60**	**54.48**
Occlusion	44.33	45.45	47.79	48.13	**53.02**	**54.55**
Whole	69.69	72.97	74.28	78.17	**77.23**	**80.45**

Occlusion and Resolution. We also test IAN using low-resolution query persons and partially occluded persons. The gallery size is fixed as 100, and several methods are evaluated on these subsets. The results are shown in Table 2.5. It is observed that all the methods perform significantly worse on both the occlusion and low-resolution subsets than on the whole test set. Nevertheless, IAN consistently outperforms E2E-PS [1] significantly.

Person Search Visualization.

We also report person search result visualization in Fig.2.7. Top-5 person search matches on the CUHK-SYSU test data are reported for 3 examples. It is observed that IAN leads to the best performance with the true matching persons ranked in front.

Fig. 2.7: Three set of examples for the top-5 person search matches on the CUHK-SYSU test data, rows 1, 4, 7 are results of the E2E-PS [1], rows 2, 5, 8 are results of E2E-PS [1] when ResNet-101 is used; rows 3, 6, 9 are results of IAN. The red box region in the first column is the probe image. The green boxes in other columns are searching results, where red boxes are ground truth results. (Best viewed zoomed-in, in color.)

Table 2.6: Performance comparison on the PRW dataset with the state-of-the-art.

Method	DPM-Alex +IDE$_{det}$ [13]	E2E-PS [1] (ResNet-101)	IAN (ResNet-101)
mAP (%)	20.20	22.39	**23.00**
top-1 (%)	48.20	61.00	**61.85**

2.4.2 Results on PRW Dataset

The obtained results on the PRW dataset are reported in Table 2.6. Our proposed method outperforms the DPM-Alex+IDE$_{det}$ method reported in [13] with a margin around 14% top-1 accuracy. More importantly, according to [13], various ways of combining of pedestrian detection methods and re-identification methods are tested for the PRW dataset, and it is shown that DPM-Alex+IDE$_{det}$ achieves the best performance among all the combinations. On the other hand, the performance of IAN is also better than that of E2E-PS [1] and DPM-Alex+IDE$_{det}$, which demonstrates the benefits of the center loss.

2.5 Conclusions

To address challenging issues in modern person search framework, we proposed a novel Individual Aggregation Network (IAN) model that can accurately localize pedestrians and meanwhile minimize intra-person variations over feature representations. In particular, we built the IAN upon the state-of-the-art object detection framework, i.e., faster R-CNN model, so that high-quality region proposal for pedestrians are produced in an online manner for person search. In addition, IAN incorporates a novel center loss which is demonstrated to be effective at relieving the negative effect caused by a large variance of the visual appearance of the same person. Meanwhile, we also performed neural network compatibility study for center loss, and we explained why dropout is not compatible with center loss. Finally, extensive experiments on two benchmarks, i.e., CUHK-SYSU and PRW, show that IAN achieves the state-of-the-art performance on both datasets, and well demonstrate the superiority of the proposed IAN network.

Based on our work in this chapter, we make a conclusion that combining the classification loss (i.e., softmax loss) with the center loss is suitable in the task of object search, such as person search. Meanwhile, it is noticed that to train the faster R-CNN framework for the task of person search, different candidate boxes should be treated differently. Only features of ground truth pedestrian boxes should be input into the center loss layer,

while features of both ground truth pedestrian boxes and RPN boxes should be input to the classification loss. We believe this is the key reason that our model leads to better performance than previous works [1] and [2]. Because center loss needs to track the feature centers of all classes, one limitation of the proposed IAN is its large GPU memory requirement, which remains to be solved in the future work.

Chapter 3

Siamese Network Ensemble for Visual Tracking

In the previous chapter, we have investigated the importance of feature learning in person re-identification in the video surveillance system. While in the following chapters, as tracking is used to localize the person who has been identified in the video surveillance system, we will investigate the feature/ model updating problems in visual tracking. This chapter starts by introducing the motivation behind this work, then we describe the network structure and loss function in a standard siamese tracker and introduce our proposed update mechanism. Afterward, the implementation detail is presented, specifying the training settings and datasets. Finally, we compare the performance of our proposed method with other SOTA trackers, and the experimental results are presented.

3.1 Motivation

The major task of object tracking in computer vision is to estimate the trajectory of a target in a video sequence. The illumination variation, occlusion, rotation, camera motion, and deformation are still the challenges for visual object tracking tasks.

Siamese instance search tracker [4] proposes an ideal matching function for visual tracking task which can handle these problems. The goal of SINT [4] is to learn a generically applicable matching function from the annotated video dataset, which is sufficiently large to model the invariance factors of different videos. Once the matching function has been trained on the external video dataset, the matching function will not be updated anymore during the tracking process.

SINT focuses on tracking efficiency, and it has no model updating, the combination of a different tracker, occlusion detection, and other mechanisms. The tracker simply returns the candidate region in a new frame that has the highest similarity score with the initial target in the first frame. Nevertheless, with such a simple model, experimental results

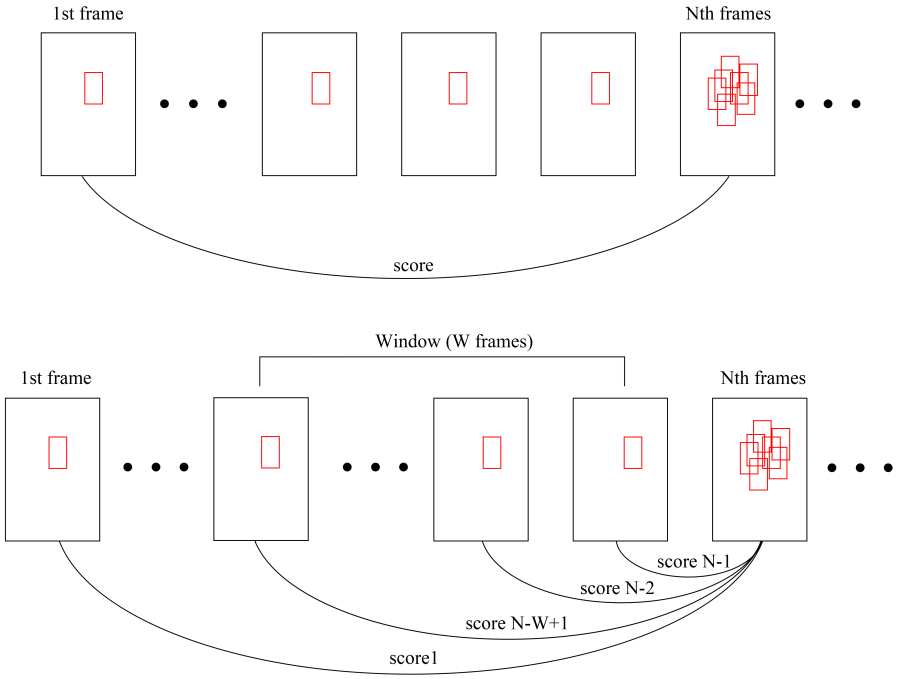


Fig. 3.1: The upper sub-figure is the matching function of SINT [4]. The bottom sub-figure is the matching function update mechanism of ours.

point out that the tracker is robust to handle the common variation of targets. Meanwhile, the matching function can be used to track unseen targets without being updated. The SINT can lead to comparable performance with existing tracking methods.

However, the matching function of SINT focuses on learning generically applicable invariance factors from different videos. For targets with large variations such as illumination variation and scale change, SINT fails to track such targets. This is because the matching function only compares the new frame regions with the initial target in the first frame; the similarity scores of the large variation objects are low, which usually leads to tracking failure.

Our work focuses on mitigating this problem. In this chapter, we propose an Ensemble Siamese Tracker (EST), where the final similarity score is also affected by the similarity with tracking results in recent frames instead of solely considering the first frame. More specifically, the tracking results in 25 recent frames are used to adjust the model for a continuous target change. As shown in Fig.3.1, the upper sub-figure of the figure shows the matching process of SINT. Candidate regions in the N-th frame are only compared with the first frame initial target. The bottom sub-figure is the matching function update mechanism of ours. Candidate regions in the N-th frame will be compared with the first frame target, and tracking results in W recent frames to adjust the final similarity score.

3.2 Proposed method

In this section, we first introduce the network architecture and loss function design of our proposed method. Then we detail the tracking inference and introduce optical flow to eliminate motion inconsistent sampling candidates.

3.2.1 Network Architecture

The two-stream Siamese network structure is adopted in our proposed ensemble Siamese tracker. Each network stream uses the VGGNet [99] structure, the same structure with the standard siamese tracker SINT [4].

The two-stream siamese network adopts very few numbers of max-pooling layers in order to achieve the goal of accurate localization. Due to the network needs to evaluate hundreds of candidate regions for each new incoming frame, it may lead to an overhead computation problem. Thus, the region pooling layer [92] is employed to process hundreds of regions. A few layers of network process the entire frame firstly. Then, the region pooling layer converts the image feature map into one fixed-length representation, which is easily proceeded for the following layers.

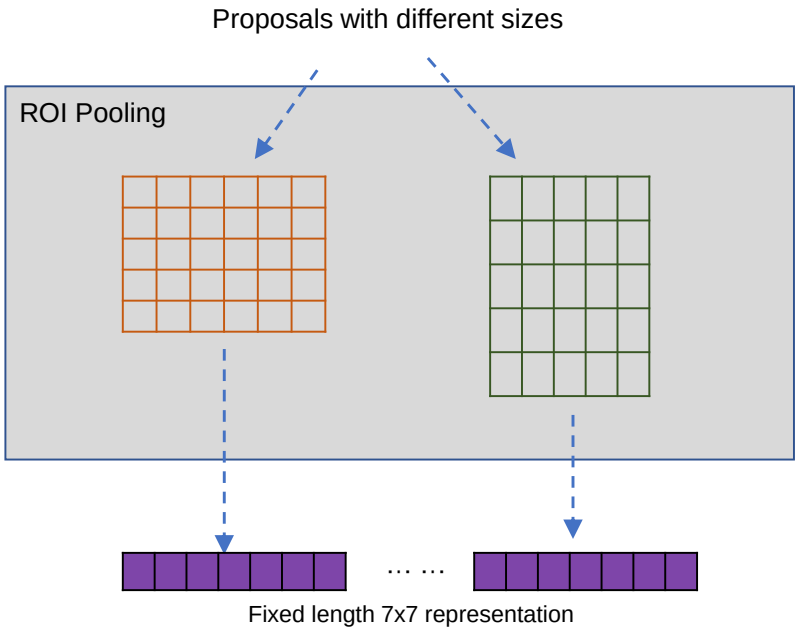


Fig. 3.2: An illusion of RoiPooling layer.

Due to the activation function of the convolutional neural network is a rectified linear unit, the output value of the network is unbounded and will vary in different scale range. Features with different scale ranges will lead to pool network representation. Therefore,

the network introduces an l_2 normalization layer before the loss layer. After the normalization layer, the different scales range features are converted into the same boundary.

The overall structure of the siamese network is shown in Fig. 3.3. 'conv', 'max pool', 'roipool' and 'fc' stand for convolution, max pooling, region-of-interest pooling and fully connected layers respectively. Numbers in square brackets are kernel size, number of outputs, and stride. The fully connected layer has 4096 units. All conv layers are followed by rectified linear units (ReLU).

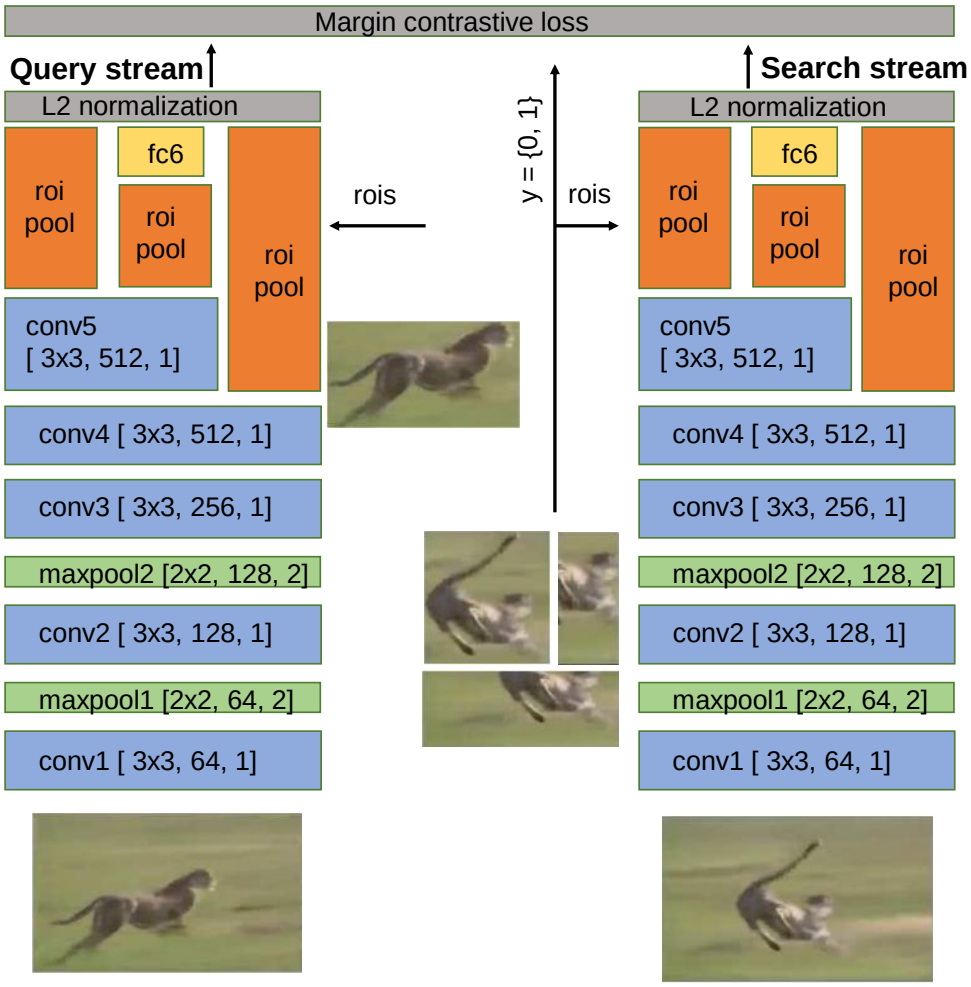


Fig. 3.3: The structure of the siamese network to learn the generic matching function for tracking. 'conv', 'maxpool', 'roipool', and 'fc' stand for convolution, max pooling, region-of-interest pooling and fully connected layers respectively. Numbers in square brackets are kernel size, number of outputs, and stride. The fully connected layer has4096 units. All conv layers are followed by rectified linear units (ReLU).

3.2.2 Network Input

The training data of EST is from the video sequence frames with the corresponding ground-truth. One of network stream is the query stream, and the other one is search

stream. For query stream, one frame is randomly picked from the video sequence that the target has an annotated location. The aim of random selection is to train network robust against many types of variation challenges. For the search stream, another frame is randomly selected in the same video, which does not need to be contiguous with the query frame. For the selected search frame, network samples on it. All candidate samples will compare with the ground-truth of the query frame to calculate overlaps. If the overlap ratio is bigger than the threshold, it is positive. If the overlap ratio is smaller than the threshold, it is negative. Then, the samples from the positive and negative pairs are used for the training process.

3.2.3 Training and Objective

From every two frames in a video, we generate multiple pairs of samples. One element in a pair is the ground truth bounding box in one frame, and the other element is a box sampled in the other frame. The pair is considered to be positive if the sampled box has an intersection-over-union overlap larger than 0.7 with the corresponding ground truth box and considered to be negative if the overlap is smaller than 0.5. The training pairs and validation pairs are generated from different videos, and therefore from different objects.

The loss layer is the last stage of our proposed method. The tracker tries to generate feature representations that are close for positive pairs, and far for negative pairs, with at least a minimum distance. The margin contrastive loss [100] is applied in this network.

$$L(x_j, x_k, y_{j,k}) = \frac{1}{2} y_{j,k} D^2 + \frac{1}{2}(1 - y_{j,k}) max(0, \epsilon - D^2) \tag{3.1}$$

The $D = \|f(x_j) - f(x_k)\|_2$ is the Euclidean distance of two l_2 normalized latent representations, where $y_{j,k} \in \{0, 1\}$. If $y_{j,k} = 1$, x_j and x_k are the same object. The ϵ is the minimum distance margin that two different objects should satisfy.

3.2.4 Tracking Inference

After the training process, the matching function of SINT [4] will not be updated. The motivation is that for object tracking, the first frame target with the annotation is the most reliable and important data. All of the candidate regions are passed through the network and compared with the first frame ground truth target. Finally, the model returns the best matching candidate region.

$$\hat{x_t} = arg_{x_{j,k}} \max M(x_{t_0}, x_{j,t}) \tag{3.2}$$

Here $x_{j,t}$ are all of the candidate samples in frame t, x_{t_0} is the ground truth target in

the first frame, and M is the matching function. The radius sampling method [101] is employed to sample the candidates. The method will sample the different scale boxes around the predicted location of the previous frame with different radiuses.

The performance of the standard siamese tracker without an update is outstanding, and the tracking speed is very fast. The matching function remains fixed after finishing the offline training stage. However, the missing updating mechanism is the key problem for these kinds of trackers, such as SINT, because, without the appearance update, the model would not be robust for sequences that with deformation, occlusion, and background clutter. The video content changes constantly; the initial object in the first frame will vary a lot after several frames.

In this chapter, we propose a matching function update mechanism to combine the first frame data and recent tracking results to adjust the matching strategy. The goal of our update mechanism is to solve the low accuracy problem for objects with large variations.

With the update mechanism, the final similarity score has two parts as show in (3.3). The regions $x_{j,t}$ are all of the candidate regions in the current frame x_t. x_{t_0} is the first frame region. x_W are the tracking results of previous W frames. M is the matching function, and λ is the weight parameter.

$$\hat{x_t} = arg_{x_{j,k}} \max(\lambda M(x_{t_0}, x_{j,t}) + (1-\lambda)\hat{M}(x_W, x_{j,t})) \tag{3.3}$$

$$\hat{M}(x_W, x_{j,t}) = \frac{1}{W} \sum_{k=t-1}^{k=t-W} M(x_k, x_{j,t}) \tag{3.4}$$

In matching function M, candidate regions $x_{j,t}$ are compared with the ground truth target of the first frame, and similarity scores of all candidate regions are obtained. Similarly, regions $x_{j,t}$ are compared with tracking results of recent frames (from $t-1$ to $t-W$) as shown in (3.4). Then, the similarity scores of W recent frames are summed up to calculate the average score. λ adjusts the weight of M and $\hat{M}$. Meanwhile, using the average scores of recent W frames is helpful to reduce the time complexity of the tracker.

We proposed update mechanism in EST uses tracking results of the latest W frames to adjust matching function, so it is robust against target variation.

3.2.5 Optical Flow

In this part, we add the Large Displacement Optical Flow (LDOF) method [102] to our proposed EST, and it further improved the tracking performance. The new tracker is named EST+.

In EST+, an adaptive candidate sampling strategy, which is mentioned in [103], is applied to help adjust the sampling range. The sampling range is set to be $30/512 * w$ to

adapt the resolution of incoming frames. The w is the width of the frames. The LDOF method is employed to generate the optical flow dataset for each video in order to help EST+ to eliminate motion inconsistent sampling candidates.

$$S = \lambda M(x_{t_0}, x_{j,k}) + (1 - \lambda)\hat{M}(x_W, x_{j,t}) \tag{3.5}$$

$$x_{c,t} = R(S)[end - c + 1, end] \tag{3.6}$$

In EST, we only select the candidate bounding box which has the maximum similarity score according to matching function (3.3). In EST+, we select the c largest-scores candidate bounding boxes in order to do optical flow matching. The function (3.5) returns similarity scores of all candidates for current frame t. The function (3.6) ranks similarity scores from small to large and picks out top c candidates $x_{c,t}$.

$$f_t = OF(F_t, F_{t-1}) \tag{3.7}$$

We employ large displacement optical flow method [102], which shows in the function 3.7, to generate optical flow dataset. The F_t is the frame image t, and f_t is the motion estimation of frame t.

$$P(x_{c,t}) = O(f_t, x_{c,t}) \tag{3.8}$$

The O in 3.8 is the function to calculate total overlap pixels between motion estimation f_t of frame t and top c candidates $x_{c,t}$. The $P(x_{c,t})$ is the total overlapping pixels number of each top c candidates.

$$Pass(x_{c,t}) = P(x_{c,t}) > P(x_{t-1}) \times \theta \tag{3.9}$$

We know the total pixels number $P(x_{t-1})$ whose location is in the previously predicted bounding box (width $\times$ height). The function (3.8) combines optical flow motion estimation data to return overlapping pixels number of each top c candidates. According to function (3.9), EST+ will only keep the candidate samples whose overlapping pixels number is bigger than $P(x_{t-1}) \times \theta$. Here θ is the lowest overlapping threshold. The candidates' samples in which overlapping pixels number is less than $P(x_{t-1}) \times \theta$ will be eliminated. These eliminated candidates are considered as motion inconsistent samples.

$$\hat{x_t} = arg_{x_{j,k}} \max(Pass(x_{c,t})) \tag{3.10}$$

If the reserved candidates $Pass(x_{c,t})$ is not null, the highest score candidate will be selected as the best matching region. If $Pass(x_{c,t})$ is null, which means all top c can-

didates are motion inconsistent samples, EST+ will reselect the highest score candidate according function (3.3).

3.3 Experiments

3.3.1 Implementation Details

Sampling of candidate boxes The radius sampling strategy [101] is used to obtain the candidate boxes. The search radius is the longer axis of the first frame ground-truth. 10 angular and 10 radial change versions are used in the experiment. For each sample location, three scales are applied on initial box location. The scale factors are $\frac{2}{\sqrt{2}}$,1,$\sqrt{2}$

Network training The ALOV [104] dataset is employed for training and validation. The selected videos in the ALOV contain different types of variations. Meanwhile, the 12 ALOV videos that are also in the visual tracking benchmark (OTB) [3] are eliminated in order to ensure the evaluation precision for OTB. Every two frames of one video will generate many pairs. There are two parts to each pair. One part is the ground-truth location of one frame, and the other one is the generated candidate samples in the other frame. Once the intersection area overlap ratio is bigger than 0.7, the current pair will be considered as positive pair. If the overlap ration is lower than 0.5, the current pair will be treated as negative pair. The generated pairs of training and validation are from different videos, which avoids confusion and increases robustness.

For this experiment, the pre-trained parameters of the network, which is trained by ImageNet classification, are used to fine-tune the Siamese network. The tuning process is stopped once the loss of validation does not decrease.

3.3.2 Optimization

Parameter setting In this project, there are two critical parameters to adjust the performance of EST. According to the matching function (3.3), the parameter λ sets the percentage of the similarity scores, which compare with the first frame, and 1-λ sets the percentage of similarity scores which compare with the recent W frames. After multiple experiments, we obtain the best combination numbers of λ and W to adjust the EST performance to the best. Table 3.1 lists out a small portion of average overlap ratio experiment results on OTB50 [3] with different combinations of λ and W. The average overlap ratio achieves the highest number once λ = 0.55 and W = 25.

Table 3.1: The average overlap ratio results of EST+ with different combinations of parameters λ and W on OTB 50 [3].

	W=25	W=30	W=35	W=40	W=45
λ=0.55	0.6512	0.6380	0.6501	0.6456	0.6573
λ=0.60	0.6505	0.6526	0.6520	0.6643	0.6539
λ=0.65	0.6499	0.6540	0.6636	0.6615	0.6535
λ=0.70	0.6642	0.6507	**0.6697**	0.6592	0.6474
λ=0.75	0.6537	0.6459	0.6600	0.6549	0.6491
λ=0.80	0.6531	0.6527	0.6518	0.6558	0.6543

3.3.3 State-of-the-art Comparison

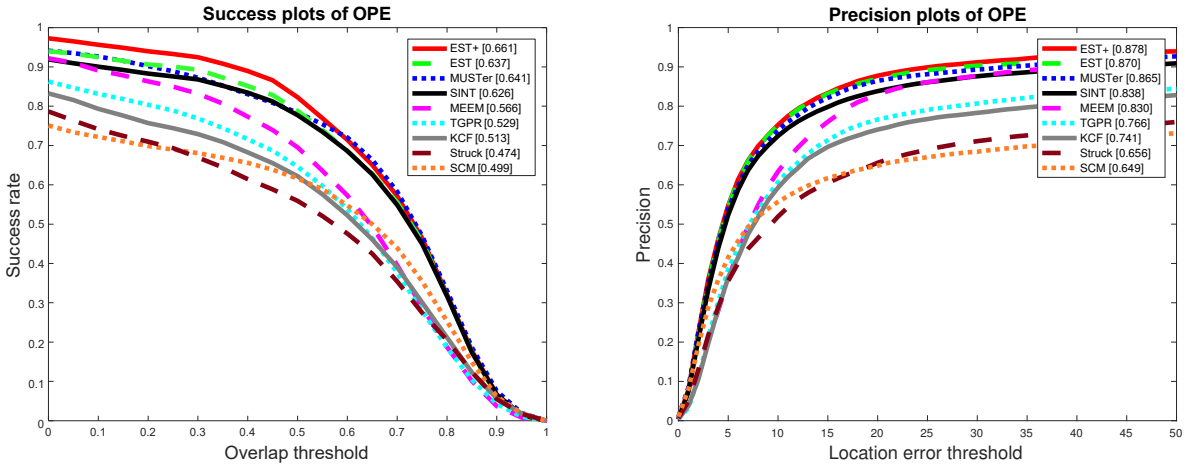
Comparison with other trackers There are 29 trackers in the OTB [3], such as some famous trackers SCM [105] and Struck [101]. This experiment also includes some recent trackers, which are used to compare with our trackers. Such as MUSTer [106], TGPR [107], KCFDP [108], and MEEM [109].

3.3.4 Dataset and Evaluation Criterion

Dataset The online tracking benchmark (OTB) [3] is used to evaluate the performance of tracker. The OTB includes many challenging factors in videos. Such as deformation, occlusion, background clutter, and motion blur, and so on. The OTB 50 and OTB 100, which have 50 videos and 100 videos, respectively, are employed to do the evaluation.

Evaluation Criterion There are two standards of evaluation criterion for OTB evaluation protocol [3]. One is the success plot and the other one is the precision plot. The results of the two criteria are the percentage of successfully tracked frames. For the success plot, one frame is treated as a successfully tracked frame once the intersection area of the predicated bounding box and corresponding ground-truth box exceeds a specified threshold. For precision plot, one frame is treated as a successfully tracked frame once the distance between the center of the predicted bounding box and corresponding ground-truth box is lower than a specified threshold. The area under curve (AUC) score is used to rank different tracking algorithms in the success plot. The precision at threshold 20 (Prec@20) is used to rank the different tracking algorithms in the precision plot.

The performance of all mentioned trackers is showed in the success plot and precision plot. Figure 3.4 points out that the performance of SINT has reached the state-of-the-art standard. The result of ensemble Siamese tracker (EST) is more accurate than the SINT method and very close to the performance of MUSTer [106], and the performance of EST+ is much better than MUSTer, and the performance is the best among all trackers. Figure 3.5 and 3.6 report the performance comparison on 11 attributes with OTB 50 [3] under AUC score and Prec@20 score. These 11 attributes are "illumination variation" (IV), "scale variation" (SV), "occlusion" (OCC), "deformation" (DEF), "motion blur" (MB), "fast motion" (FM), "in-plane rotation" (IPR), "out-of-plane rotation" (OPR), "out-of-view" (OV), "background clutter" (BC) and "low resolution" (LR). The results of EST are the best in SV, OPR, IPR, IV, MB, BC, and FM attributes. For the conditions of low resolution and motion blur, the performance of EST is much better than MUSTer [106]. Due to the matching function updating mechanism, the performance of EST on OPR, SV, IPR, IV, OCC, and BC is much better than SINT [4]. The updated matching function of EST can adapt to more challenging conditions. Especially, the performance improvement of EST+ on SV, OCC, MB, OV, and LR is remarkable.



(a) Success Plot (b) Precision Plot

Fig. 3.4: The comparison results on OTB 50 [3]. The improvement of EST and EST+ over SINT is clear, and EST has reached state-of-the-art performance. The curve of EST+ is the highest among all trackers.

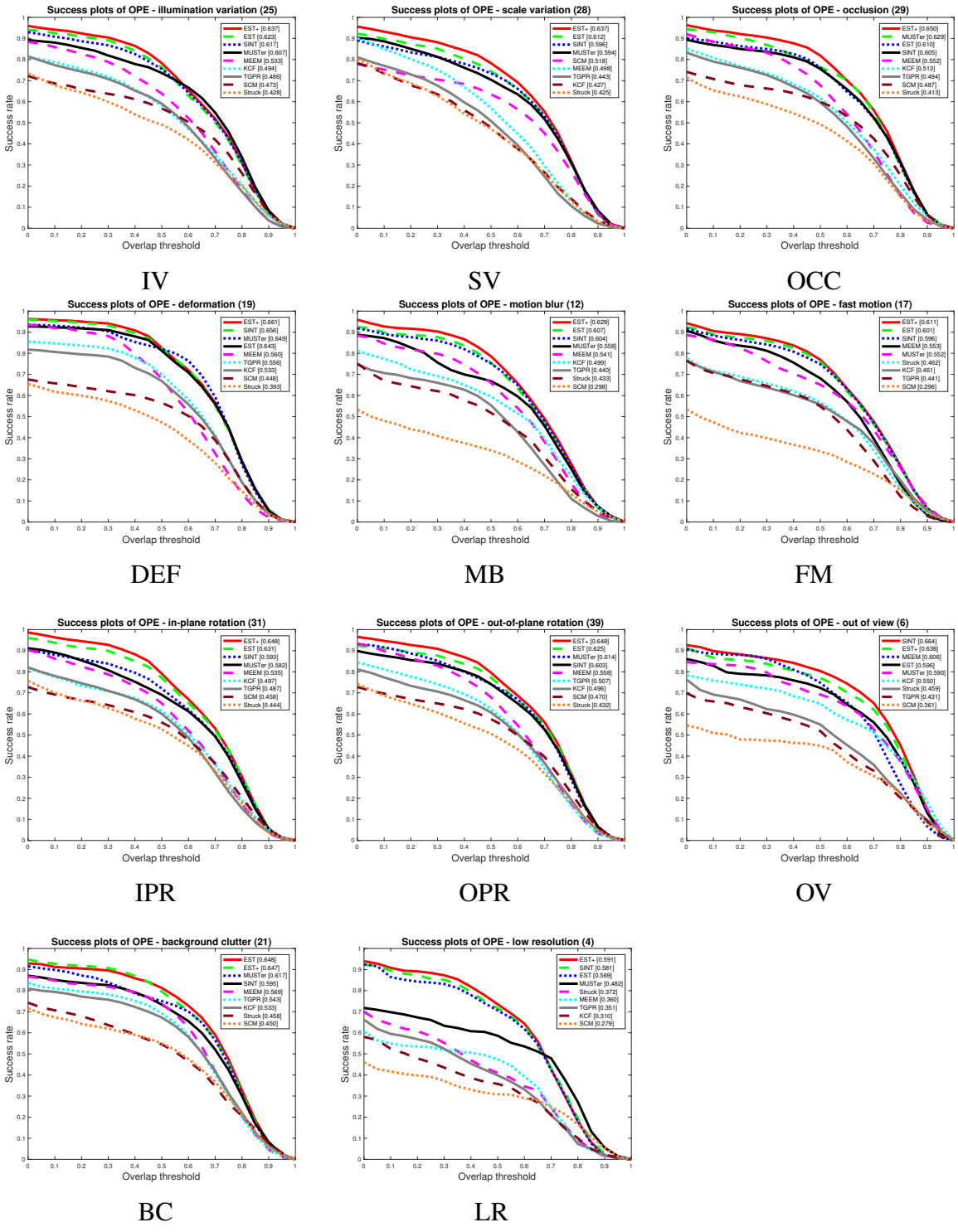
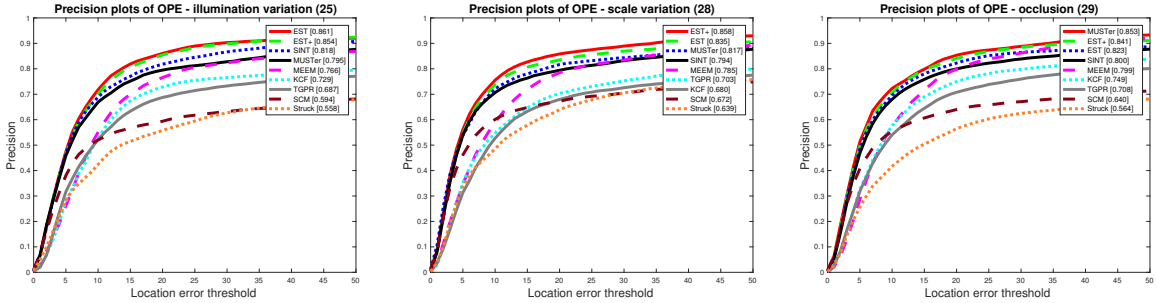


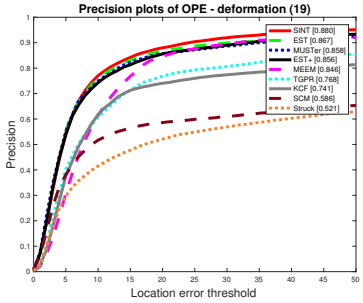
Fig. 3.5: The performance of 8 trackers for 11 attributes on OTB50 [3] under AUC score. Only the performance of EST in DEF, LR, and OV attributes is lower than SINT [4]. For the attributes IV, OPR, SV, MB, FM, IPR, and BC, the performance of EST is better. For EST+, only the OV attribute is lower than SINT.



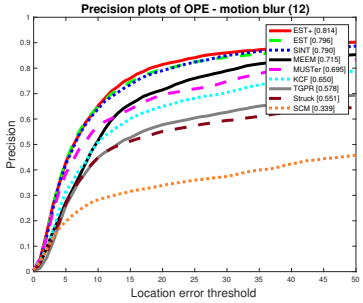
IV

SV

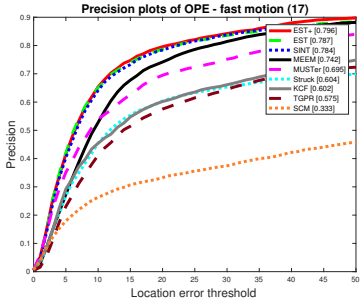
OCC



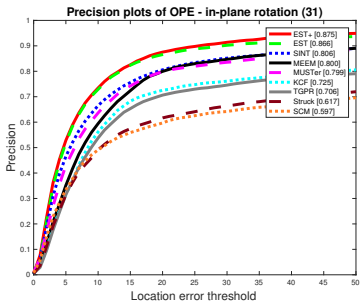
DEF



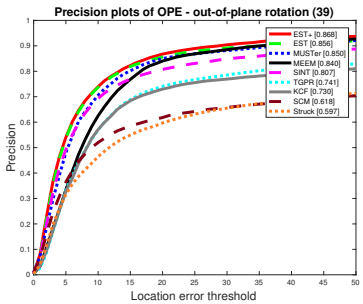
MB



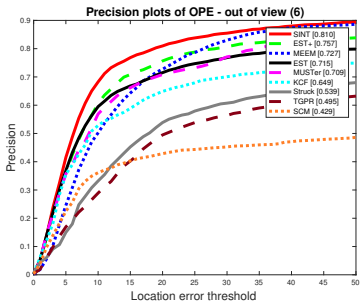
FM



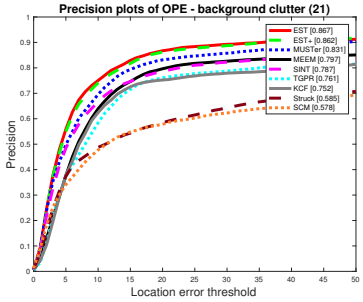
IPR



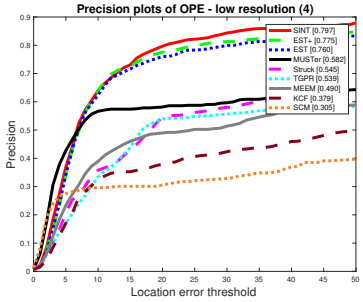
OPR



OV



BC

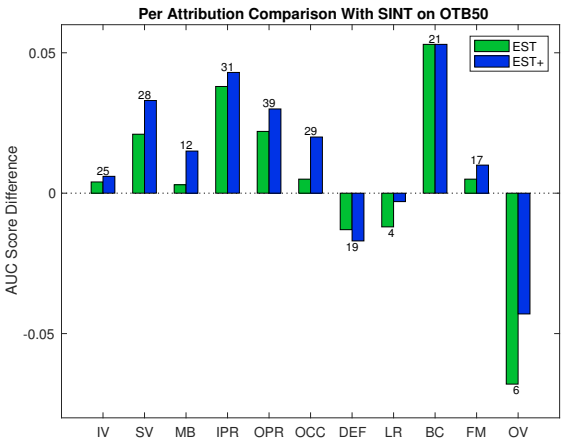


LR

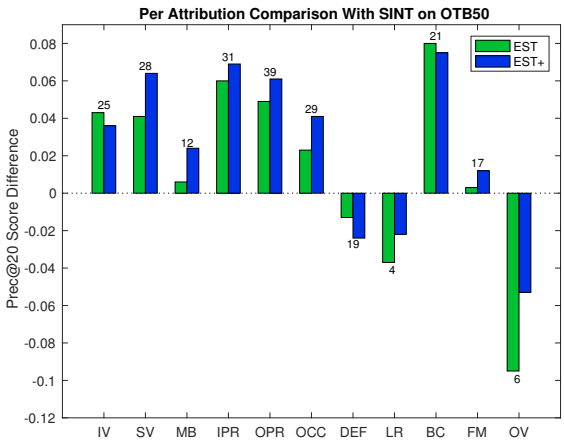
Fig. 3.6: The performance of 8 trackers for 11 attributes on OTB50 [3] under Prec@20 score. Only the performance of EST in DEF, LR, and OV attributes is lower than SINT [4]. For the attributes IV, OPR, SV, MB, FM, IPR, and BC, the performance of EST and EST+ is better.

Per Attribution Comparison The 11 attributions in OTB [3] videos include fast motion, low resolution, deformation and so on. We evaluate the performance of EST on these 11 attributions and compare them with SINT [4] and MUSTer [106] in order to obtained more detailed comparison data. Figure 3.7 shows the comparison results on AUC and Prec@20 scores. From the results of figure 3.7, SINT only performs better than EST in deformation, low resolution, and out-of-view. For the performance of MUSTer [106], only the results of deformation and occlusion attributions are better than EST. However, the improvement of EST is obvious. Furthermore, the adaptive candidate sampling strategy and large displacement optical flow method are helpfully for EST. The performance of EST+ is much better than EST in almost all of the attributions. When some portion of the target leaves the view, incorrect tracking results are used to update the matching function. While the original no-update tracker would not be affected under such situations, a judgment mechanism, such as a decision unit, can be developed to avoid erroneous updates, which will be discussed in Chapter 4. The improvement on Prec@20 score is competitive. The MUSTer [106] is mainly better in ”occlusion”, and the SINT [4] is mainly better in ”out-of-view”. Whereas, the EST is much better in ”illumination variation”, ”scale variation”, ”motion blur”, ”fast motion”, ”in-plane rotation”, ”out-of-plane rotation” and ”background clutter”.

On the other hand, the proposed tracker is also evaluated on OTB 100 [3]. One hundred videos include various visual tacking challenges. The results of OTB 100 [3] video evaluation provides a more comprehensive description of the performance of trackers. For the 100 videos evaluation, we used four other trackers to compare with our trackers. The other 4 trackers are KCFDP [108], TGPR [107], SINT [4], and MEEM [109]. In figure 3.10, the performance of all mentioned trackers are shown in the success plot and precision plot. The performance of our trackers is the best in this experiment. The accuracy is much better than SINT [4]. We also evaluate the 11 challenging attributes performance of EST+ and EST on OTB100 [3]. In figure 3.11, the results point out that the low resolution (LR) attribute data of EST and EST+ has exceeded SINT [4]. The SINT is only better in ”out-of-view” and ”deformation.”

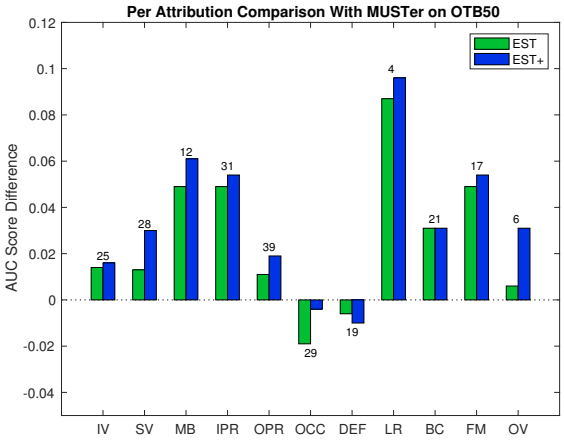


AUC Score

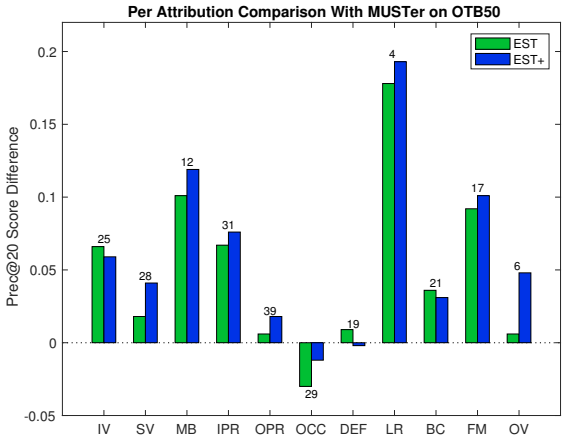


Prec@20 Score

EST+, EST and SINT



AUC Score



Prec@20 Score

EST+, EST and MUSTer

Fig. 3.7: Per attribution comparison of EST and EST+ with SINT [4] and MUSTer [106] on AUC and Prec@20 scores. The bars stand for the AUC and Prec@20 score differences between trackers. The positive bar means the performance of EST and EST+ is better than SINT and MUSTer. The numbers on the top of each bar mean the total number of the videos that have the same attribution in OTB50.

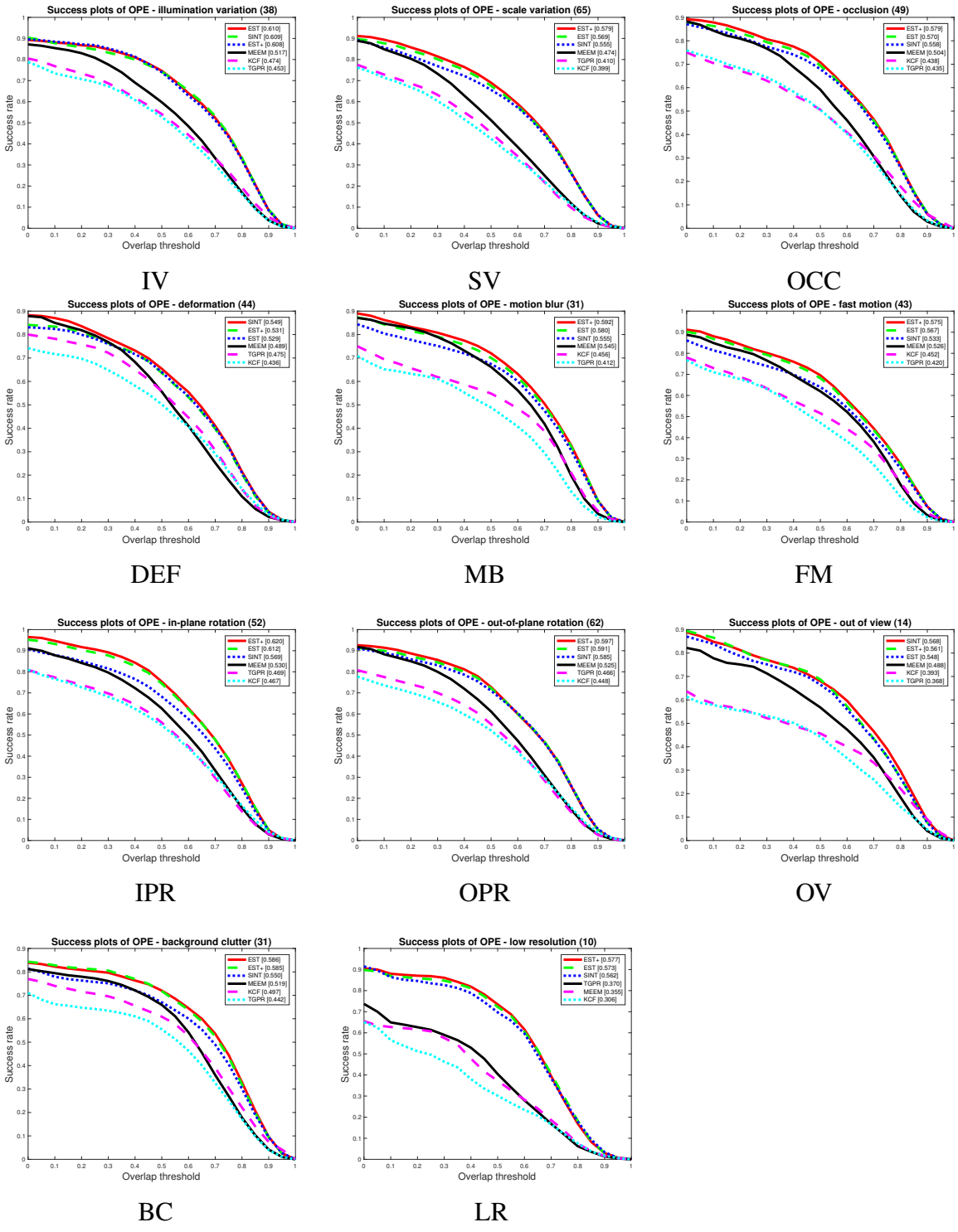


Fig. 3.8: The performance of 8 trackers for 11 attributes on OTB100 [3] under AUC score. Only the performance of EST in DEF and OV attributes is lower than SINT [4]. For the attributes IV, OPR, SV, MB, FM, IPR, LR, and BC, the performance of EST and EST+ is better.

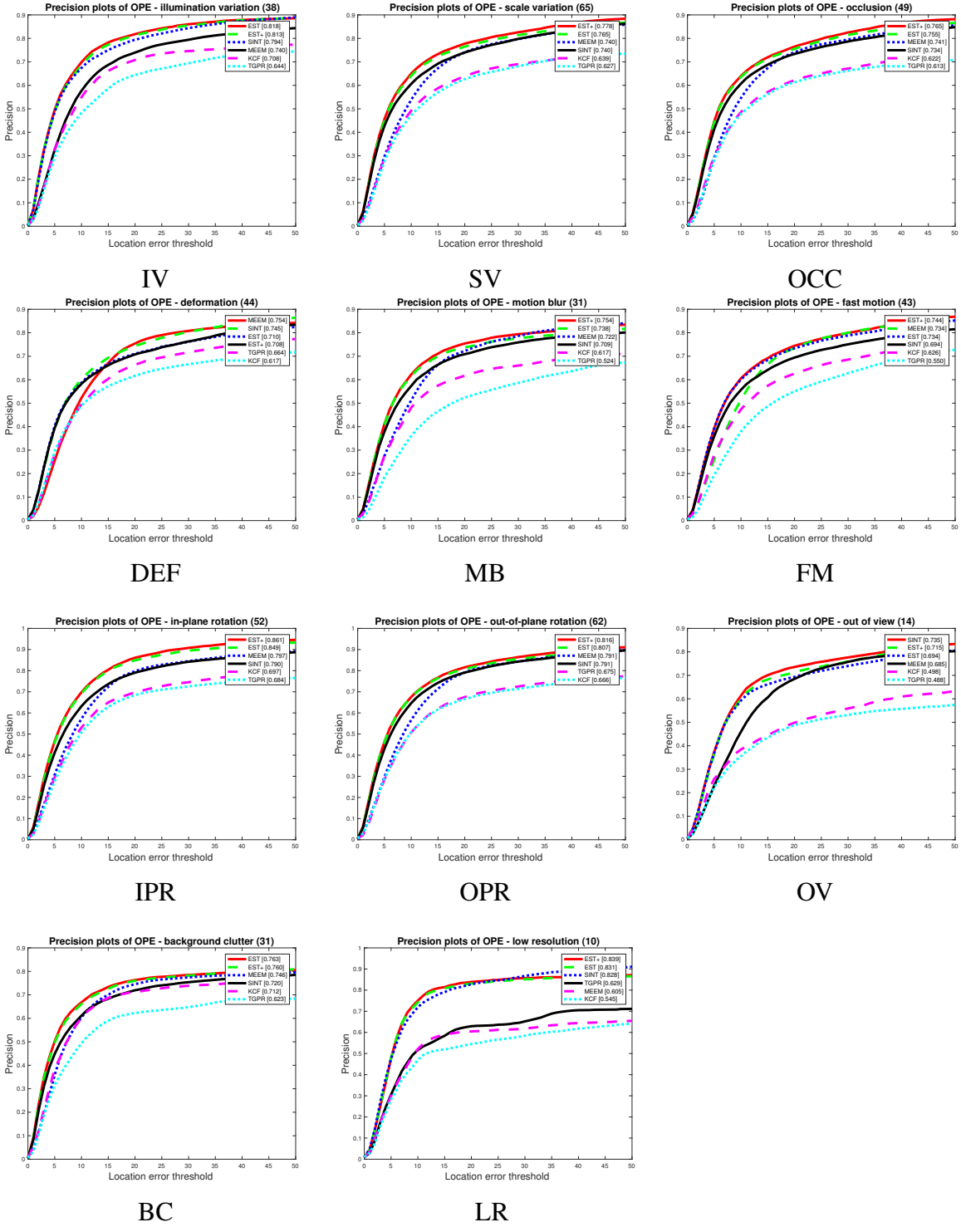
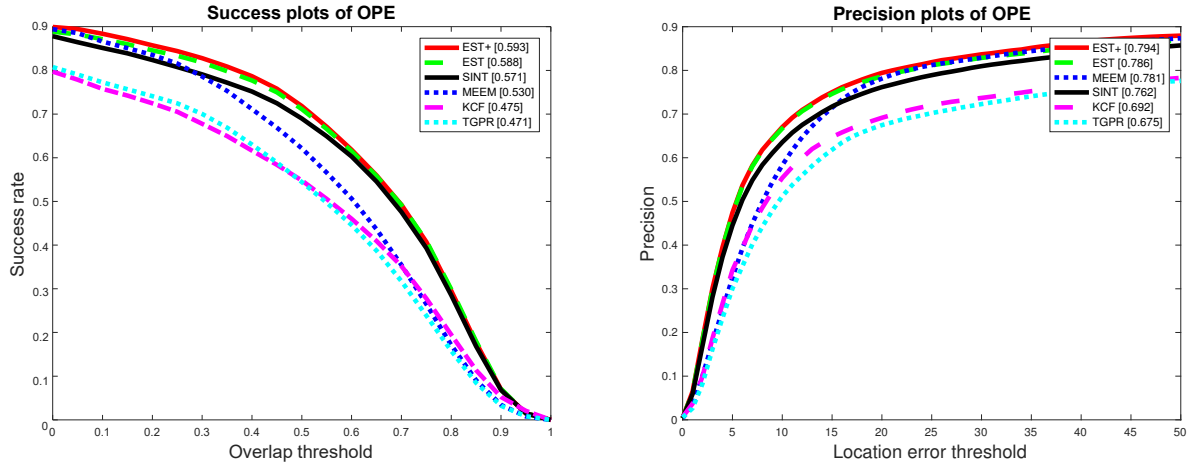
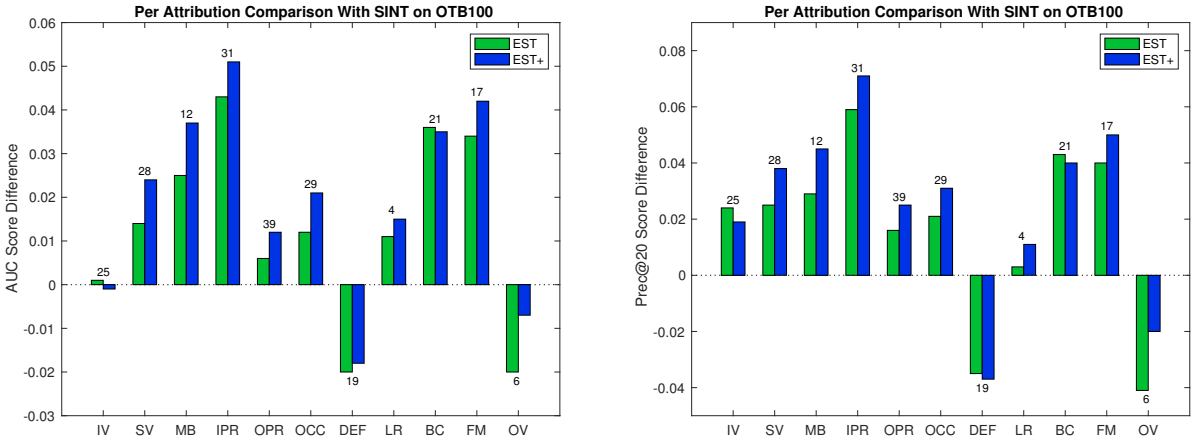


Fig. 3.9: The performance of 8 trackers for 11 attributes on OTB100 [3] under Prec score. Only the performance of EST in DEF and OV attributes is lower than SINT [4]. For the attributes IV, OPR, SV, MB, FM, IPR, LR, and BC, the performance of EST and EST+ is better.



(a) Success Plot (b) Precision Plot

Fig. 3.10: The performance of five recent trackers is compared on OTB 100 [3]. The improvement of EST+ and compared with SINT is remarkable.



AUC Score Prec@20 Score

EST+, EST and SINT

Fig. 3.11: Per attribution comparison of EST and EST+ with SINT [4] on OTB100 [3] under AUC and Prec@20 scores. The bars stand for the AUC and Prec@20 score differences between trackers. The positive bar means the performance of EST and EST+ is better than SINT and MUSTer. The numbers on the top of each bar mean the total number of the videos that have the same attribution in OTB100.

Overlap ratio comparison The overlap ratio comparison for EST is also evaluated on OTB 50 and OTB 100 [3]. The results are reported in table 3.2. In both datasets, EST+ and EST achieves higher overlap ratio than SINT and the gain is more than 0.02 for OTB 100. The overlap area is one meaningful evaluation criterion of tracking accuracy. The average ratio is obtained from the ratio of intersection area between ground-truth and

predict box.

Table 3.2: The average overlap ratio for SINT [4] and EST on OTB 50 and OTB 100 [3].

	OTB50	OTB100
EST+	0.6697	0.6006
EST	0.6459	0.5950
SINT	0.6342	0.5780

3.4 Conclusions

In this chapter, we propose a novel online matching function updating mechanism to adjust the similarity score against object variation. Our proposed ensemble Siamese tracker not only uses the first frame information but also consider the tracking results of adjacent frames to update the matching model. This mechanism enables EST to learn the target variation information and build a strong feature representation for tracking. Therefore, the EST is more robust against deformation, occlusion, and other variation challenges. Our proposed updating mechanism has been evaluated on a visual object tracking benchmark. The performance of EST on most of the video challenges like fast motion, background clutter, and scale variation, is much better than other trackers. The improvement of EST is considerably compared with standard siamese trackers such as SINT, where no model updating is used.

Chapter 4

Correlation Filter Selection for Visual Tracking

In previous chapters, we discussed the feature learning in person re-identification tasks and pointed out the importance of getting discriminative features for the re-identification and tracking tasks. The correlation filter based trackers use the correlation filter to build the tracking target's appearance model, which aims at producing discriminative features. So in this chapter, we further study the feature/ model updating problem of the correlation filter based approach other than the siamese trackers, which are described in chapter 3, to provide a more discriminative feature representation for the visual tracking task. Especially, in this work, a novel model selection method for correlation filter based trackers using deep reinforcement learning is presented.

4.1 Motivation

Discriminative correlation filter (CF)-based trackers [110][60][44][111][46] achieve a good trade-off between accuracy and speed by efficiently solving a ridge regression problem in Fourier frequency domain. Regularized correlation filters [62][112] are proposed to further enhance the tracking accuracy. Gladh *et al*. introduces motion information along with hand-crafted features for CF tracking [113]. Mueller *et al*. propose a context-aware CF tracking [63]. Sophisticated learning schemes are proposed to achieve powerful feature representation [50][66].

Most discriminative model-based trackers exploit the target from a given bounding box directly, which is used to build the appearance model of the objects at the latter stages. During the tracking process, new image patches generated from new frames are supplemented to further update the CF model. Generally, a small update-rate is usually preferred for CF trackers in order to maintain model stability. These trackers may easily suffer from a drift problem, especially in challenging environments such as partial

occlusions, background clutter, and low resolution.

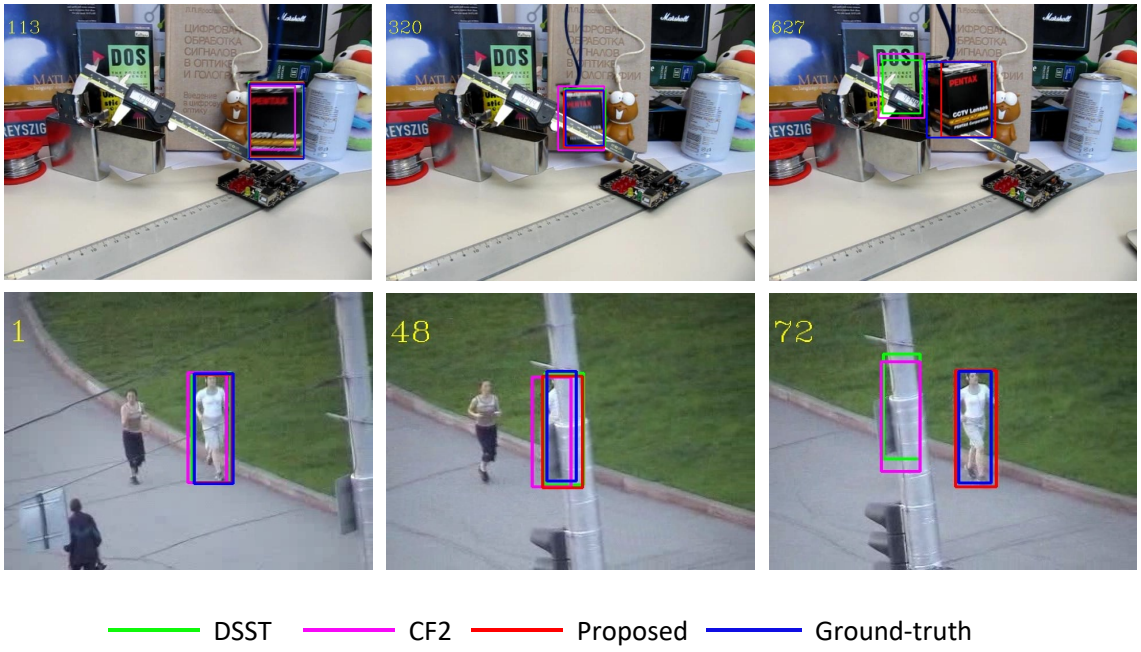


Fig. 4.1: Visualization of 3 tracking results. Green, purple, red box denote tracking results of DSST, CF2, and the proposed tracker, respectively; blue box denotes the ground-truth box of tracking sequences. During the tracking process, targets suffer from partial occlusion, while other trackers do not realize this and result in model drift. The proposed tracker with the decision unit updates the appearance model guided by the response map and skips updating if not necessary.

An example of this is illustrated in Fig. 4.1, the tracking model is initialized with a target box in the first frame, which is also used as the ground truth for subsequent analyses. A discriminative CF can easily be obtained using a two-dimensional Gaussian label whose center is the same as that of the target box. However, during the tracking process, we notice the target becomes partially occluded by foreground objects in some cases, as shown in the middle column of Fig. 4.1. However, the CF model is oblivious to this occlusion issue and kept updated without even evaluating the reliability of new image patches. Tracking results that are generated with such poorly updated CF models influence the subsequent updates of the CF model. A number of such updates accumulate the errors and results in irrecoverable model drift.

To mitigate such model drifts, Gao *et al.* proposes a deep network to learn a relative model to deal with target appearance changes [114]. Zhu *et al.* uses multiple fine-grained foreground-versus-contextual-cluster models to provide more discriminative classifications [115]. The multi-scale and Spatio-temporal context is explored by [116] to choose better tracking samples. Yao *et al.* proposes a semantics-aware method [117] to enhance the appearance model in visual object tracking. However, it is not flexible to transfer a

relative model or add semantics information into a CF-based tracker. Furthermore, such a transfer process entails a substantial investment in time towards re-modeling the relative model. Wang *et al.* proposes the structured correlation filter [118], which couple interactions between a static model and a dynamic model to handle long term tracking. A decision-making network is proposed using a Siamese tracking framework [119], which also aims to solve the model drifting problem. Base on that, we naturally consider that a selection for CF models will contribute to building a better discriminative appearance model for visual tracking.

In recent years, progress in deep learning has been influential in the domain of visual tracking. Combining CF with convolutional features has been considered in several studies [64, 65, 49, 66, 67, 120]. These studies show that deep convolutional networks (DCNs) that are pre-trained with certain large-scale data and adaptive correlation filter are complementary. The CF-embedded DCNs are shown to be able to achieve state-of-the-art performance on many object tracking benchmarks [3].

An approach of CF-based tracking reformulating the CF into a convolutional layer can offer end-to-end learning. For example, in [66], instead of solving the CF with a closed-form solution, it is learned as kernels of a convolutional layer, which can benefit from end-to-end training. In this framework, the CF is updated by back-propagation. However, despite using residual learning to enhance the feature representation, noisy updates are still a problem. Meanwhile, the application of Siamese frameworks has also been explored in visual tracking, including SiameseFC [48], DSaim [121], SINT [16] and CFNet [122]. They all employ a powerful convolutional network to address the similarity learning problem for visual tracking.

Although the utilization of both convolutional neural network (CNN) and CF have been instrumental in addressing a number of problems and in achieving rather remarkable outcomes, there are still a number of problems still remain to be addressed.

First, when obtaining discriminative features for tracking, owing to the underlying complexity of parameter models, a significant amount of computational resources are needed. In addition to this, large models tend to introduce severe over-fitting problems. Models like VGG-19 tend to be an inferior option for CF-based trackers. Other than one forward pass in the convolutional network for feature extraction, CF trackers need additional time to compute the correction filter in the Fourier frequency domain, which can hardly benefit from GPUs. Nevertheless, operating in the Fourier frequency domain speeds up CF.

Second, most existing trackers update tracking models at each frame. Especially for CF trackers, a simple moving average scheme is exploited in essence. For example, the state-of-the-art tracker ECO [50] takes the sparser update to refine their model. This may,

however, cause deterministic failures once the target is inaccurately detected, severely occluded, or totally missing in the current frame. Meanwhile, it is hard to judge whether an update for the CF is reliable or not. Therefore, a more sophisticated model update strategy is necessary to handle this issue.

Motivated by the fact that the CF model might be updated with inaccurately tracking results, some temporally old CF models might be able to generate better tracking results than the latest one. In this chapter, we propose to maintain more than one CF model. Instead of always using the latest CF model, the most suitable CF model will be selected and used to generate tracking results. To select the most suitable one among multiple models, reinforcement learning is deployed.

Convolutional features contribute to robust feature representation. Therefore, in our proposed method, we engage a light-weighted convolutional network as a feature extractor. Meanwhile, the performance of CF-based trackers, in comparison to other trackers, is a great advantage. While a standard CF solver is exploited for tracking, the net structure in [67] satisfies the need for fast convolutional feature extraction. Based on this work, we investigate the model update problem by formulating CF model updating as a Markov decision process.

Reinforcement learning has been studied for visual tracking recently [82], [83]. Huang *et al.* [82] succeeds in utilizing Q-learning [86] for shallow-level or high-level feature selection. ADnet [83] uses policy gradient learning and trains action dynamics for tracking with annotated visual tracking sequences. Recently, Dong *et al.* [123] propose to use continuous deep Q-Learning for hyperparameter selection in tracking. Our work is significantly different from these existing works, in that we are studying the model update issue with reinforcement learning.

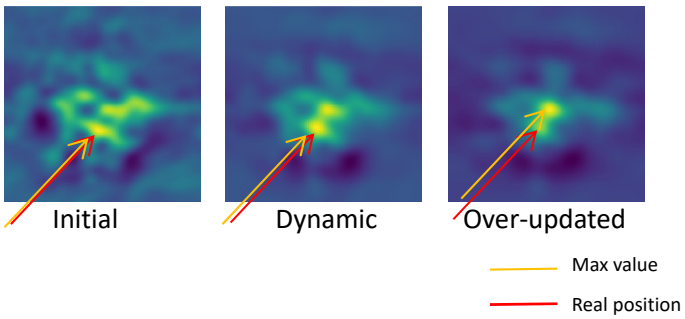


Fig. 4.2: A visualization of 3 response maps from CF models of different stages. Bright yellow color denotes regions where high probability the target will be, while the dark blue color represents a relatively low probability. After a period of the update, CF model drifts, and the over-updated model produces a good-looking response map while failing to track the true target. (Better viewed in color)

The main contributions of this work are as follows:

1. We propose a novel approach for selecting an optimal model among multiple CF models that are updated and maintained in parallel. This approach addresses a number of concerns that arise from a single CF model, such as drift;

2. We propose a reinforcement learning-based approach for optimal model selection. To the best of our knowledge, this is the first time that reinforcement learning is utilized for model selection among multiple CF models;

3. We utilize a light-weight feature extractor and proposed a small decision network so that the proposed approach can be deployed in real-time applications, where the frame rates are high;

4. We exhaustively evaluate the proposed approach on OTB100 and OTB2013 benchmarks. Our results show an average success rate of 62.3% and average precision of 81.2%. These results are better than the approaches that adopt traditional CF trackers without multiple model selection.

4.2 Our Proposed Approach

In visual tracking, the traditional CF model might be updated with inaccurately tracking results, and suffers from drift problem, as shown in Fig. 4.1. To mitigate the issue of possible inaccurate model update during the tracking process, we propose to maintain more than one CF model for visual tracking. Instead of always using the latest CF model, the most suitable CF model will be selected using reinforcement learning. More specifically, the current search frame is input into the convolutional feature extractor, and several response maps are generated utilizing all the maintained CF models. Each response map corresponds to one CF model. Different response maps at a one-time step are visualized in Fig. 4.2. The RL algorithm PPO [69] is utilized to select the most suitable CF model based on the convolutional feature of the corresponding response map. Then, the tracking bounding box is generated using the corresponding CF response map. Finally, the CF models are updated with the new tracking bounding box, which will be used for the next frame. The overall proposed framework is presented in Fig. 4.2. The pseudo-code of the algorithm is described in Algorithm 1.

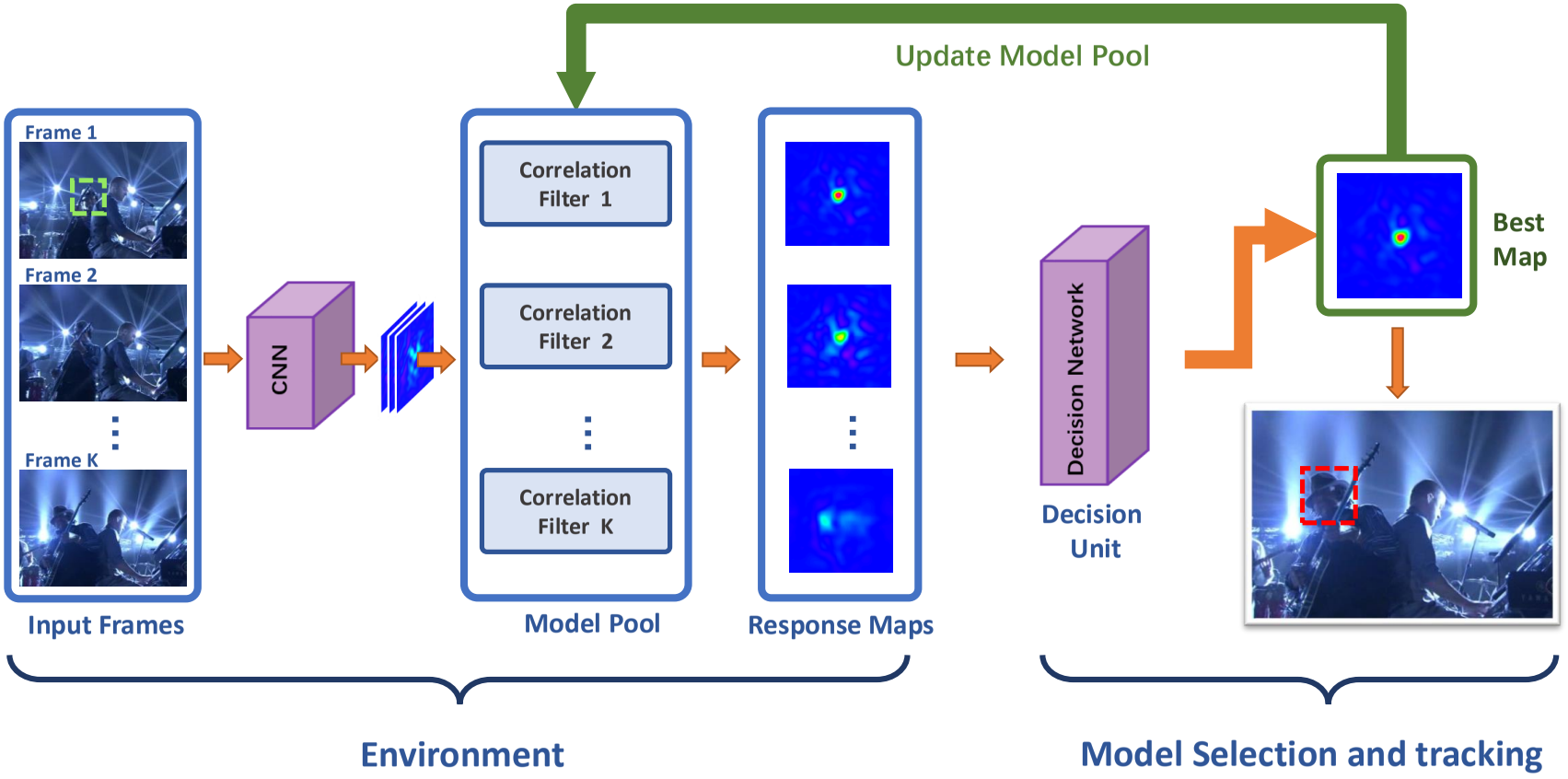


Fig. 4.3: Given a sequence containing L frames, we take the target location in the first frame and use its feature to initialize the CF model. Different states at each time-step would produce CF models sharing different memories of target appearance. From each CF model, we obtain one corresponding response map. The trained decision network will select the response map based on learned experience and finally point out the target locations.

Algorithm 1: Visual tracking with multiple CF models and reinforcement learning.

Input:
Tracking sequence of length L
Initial object location x_0
Output: Target object location in frame t x_t

Initialize CF model M_0 with the ground-truth
Set history CF model M_1, .. , M_{K-1} = M_0
for $t = 1$ *to* L **do**
  for $i = 1$ *to K* **do**
    Calculate response maps P_i with each M_i
    Produce confidence score via decision net $\pi(a_t|s_t;\theta)$ for each P_i
  end
  Choose the response map with maximum confidence P_m ;
  Localize the target according to chosen P_m;
  Update corresponding CF models and save to history;
end

4.2.1 Light-weighted Correlations Filter Model

The CF-based trackers have demonstrated strong capability on building accurate models with slight online model updating. Recently, many proposed new tracking algorithms [66, 70] benefit from the advantage of CF. A standard CF can be solved following the objective function (4.1),

$$\arg\min_{f} = ||\psi(x) * f - g||^2 + \lambda||f||^2, \tag{4.1}$$

where f is the CF, $*$ is the circular correlation or convolution operation, and ψ is a feature extractor, x is a cropped image centered on the target, and $g \in R^{H\times W}$ is the desired Gaussian shaped response map label. f can be efficiently solved by transforming (4.1) into the Fourier domain. The Fourier domain representation of f can be calculated as (4.2).

$$F = \frac{\bar{G} \odot \bar{X}}{\bar{X} \odot X + \lambda}, \tag{4.2}$$

where G is the Fourier transformation from Gaussian shaped label g, X is the Fourier transformation of x, and the bar means complex conjugation. Operator $\odot$ is the element-wise product.

New search image z around the target in the next frame is cropped with 2 to 4 times of the target size. A response map P in the Fourier domain is obtained by (4.3).

$$P = F \odot \bar{Z}, \tag{4.3}$$

where Z is the Fourier transformation of z. At a new tracking frame, once the CF F is ready, the tracking bounding box center locates at the coordinate that has the maximum

response value.

Typically, the numerator A and denominator B of the CF in (4.2) are updated separately using a moving average mechanism.

$$A_t = (1-\eta)A_{t-1} + \eta G \odot \bar{X}_t, \tag{4.4}$$

$$B_t = (1-\eta)B_{t-1} + \eta X_t \odot \bar{X}_t + \lambda, \tag{4.5}$$

Traditional CF trackers update tracking models frame by frame without considering their tracking results. This may cause an inaccurate model update when occlusion or object missing occurs. Designing a criterion to produce a high-confidence update has been exploded by [71]. Average peak-to-correlation energy (APCE) is proposed to select high-confidence response maps that effectively prevent CF model from corruption. In this work, instead of calculating an APCE score to decide whether to update the model or not, we introduce a learning algorithm to perform multiple model selection.

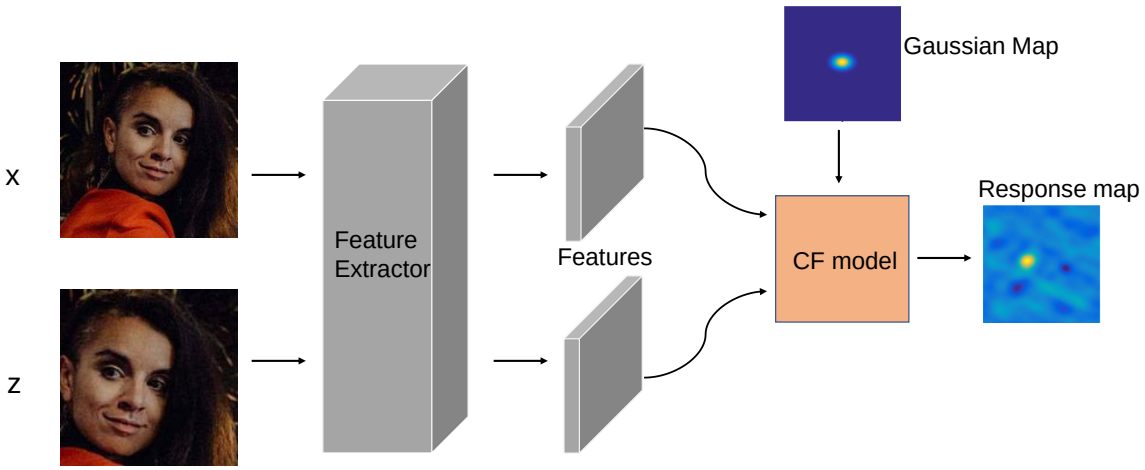


Fig. 4.4: The architecture of the CF network.

The traditional CF based trackers usually use hand-craft features or existing VGG features from a pre-trained model, while we aim to bridge the gap between feature extractors and correlation filters and tune the CF model simultaneously. Considering the tracking inference speed, the model should also be light. As shown in Fig. 4.4, the CF network is realized by cascading a feature extractor with a CF model to get the response of the object location. Giving the features of search patch, the desired response should get a high response at the real location. The objective function can be formulated as 4.6.

$$L(\theta) = ||\psi(z,\theta) * f - g||^2 + \gamma||\theta||^2, \tag{4.6}$$

$$f = \mathcal{F}^{-1}\left(\frac{\bar{g} \odot \bar{\psi}(x,\theta)}{\bar{\psi}(x,\theta) \odot \psi(x,\theta)^* + \lambda}\right), \tag{4.7}$$

where θ is the parameters in the CF network.

We can propagate the error backwards to the real-value feature maps according to method [67], and the rest backpropagations are conducted as traditional convolutional neural network optimization.

$$\frac{\partial L}{\partial \varphi(\mathbf{z})} = \mathcal{F}^{-1}\left(\frac{\partial L}{\partial\left(\hat{\varphi}(\mathbf{z})\right)^{*}}\right) \tag{4.8}$$

$$\frac{\partial L}{\partial \varphi(\mathbf{x})} = \mathcal{F}^{-1}\left(\frac{\partial L}{\partial\left(\hat{\varphi}(\mathbf{x})\right)^{*}} + \left(\frac{\partial L}{\partial \hat{\varphi}(\mathbf{x})}\right)^{*}\right) \tag{4.9}$$

where $(\cdot)^{*}$represents the complex conjugate of a complex term $(\cdot)$. In this way, we can retain the efficience of correlation filter method and training the network on large-scale datasets. The convolutional layers of the network consist of $conv1$ from the structure of VGG net, with all pooling layers removed and 32 output channels. So the model is light-weighted compared with other deep learning based trackers.

4.2.2 Model Selection Using Reinforcement Learning

After introducing the light-weighted correlation filter network, we then describe the reinforcement learning settings in model selection. We formulate object tracking as a discrete control problem, which requires the tracker to rapidly respond to the object's movement and appearance change based on CF response maps.

In the RL set-up, the agent interacts with the environment by taking an action corresponding to the current state. After the agent receives a state, the agent uses its policy to take action. Both the environment and the agent will transit to a new state based on the current state and the chosen action. A reward evaluating the made action will be used as feedback and sent to the decision unit to learn and improve the policy.

The block diagram of reinforcement learning for visual tracking is illustrated in Fig. 4.5, the decision network which consists of a policy network and a value network, takes the observation from the environment and produce instructions for the agent to act. a_t here is to select appropriate CF models that generate a response map to speculate target location. A collection is sampled after a series of actions, which is used to update the decision network.

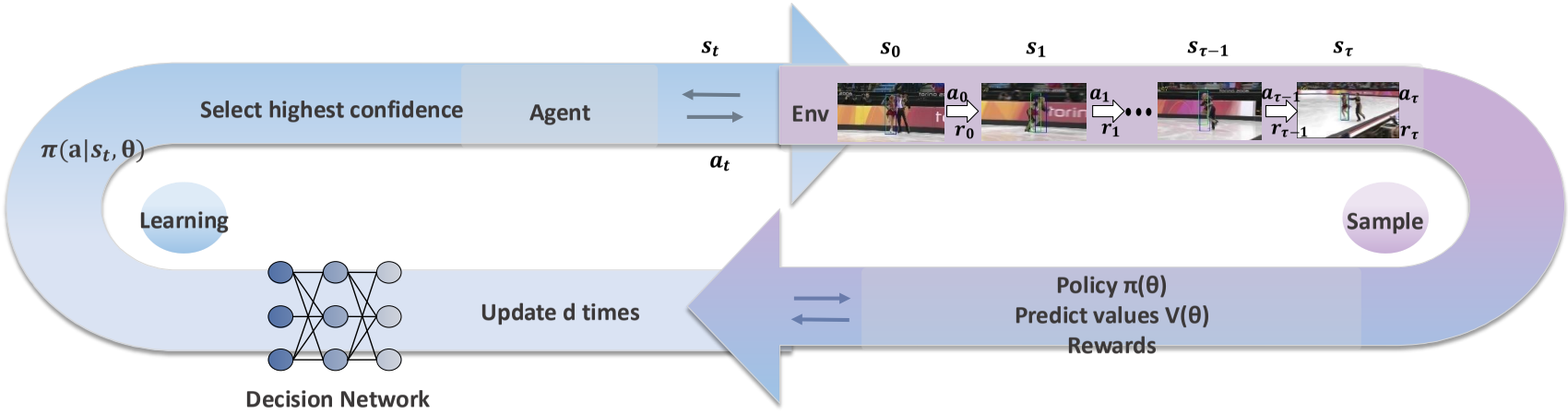


Fig. 4.5: Training process of reinforcement learning algorithm for tracking. Decision network which consists of a policy network and a value network, takes the observation from the environment and produce instructions for the agent to act. a_t here is to select appropriate CF models which generate response map to speculate target location. A collection is sampled after a series of actions, which is used to update the decision network.

At frame t, we denote the observed state by s_t, which is a set of response maps generated by the CF models. Denote the action by $\mathcal{A}$ of size k, which represents selecting k different CF models. At each frame, we draw an action a_t ($a_t \in \mathcal{A}$), from a policy distribution. Then, a reward, r_t, according to the tracking results, can be calculated and obtained after the agent's action. The reward is computed through reward function $r_t = g(s_t, a_t)$, and we will detail the function later. The old state is updated by the agent, and a new state s_{t+1} will be generated, which is an unknown state depending on the taken action. Repeating this process, we can observe a sequence of {state, action, reward}, denoted as $\tau = \{(s_0, a_0, r_0), \cdots, (s_t, a_t, r_t), \cdots, (s_T, a_T, r_T)\}$. Here, at time-step T, the tracker reaches the end of the sequence or it fails to locate position inside the image. The collected samples are used to update the decision network.

Meanwhile, we can learn policy function $\pi(s_t; \theta)$ and value function $V(s_t; \theta)$ over the trace τ with stochastic policy gradient and value function regression using PPO [69]. The loss function $L_t(\theta)$ is defined as follows, which combines the policy surrogate and value function term.

$$L_t(\theta) = min(\ Ratio * A_t,\ clip(Ratio, 1-\epsilon,\ 1+\epsilon)\ A_t), \tag{4.10}$$

where

$$Ratio = \frac{\pi(a_t|s_t;\theta)}{\pi(a_t|s_t;\theta_{old})}, \tag{4.11}$$

Here, θ_{old} is the vector of policy parameters before the update. θ is the new policy parameters. $\pi(a_t|s_t;\theta)$ is the policy function, which defines the probability to take action a_t, under the state s_t and policy parameters θ. Similarly, $\pi(a_t|s_t;\theta_{old})$ is the the probability to take action a_t, under the state s_t and the old policy parameters θ_{old}.

The clip function is defined as: given an interval, values outside the interval are clipped to the interval edges. The clipped surrogate objective limits the variation of the surrogate, which adds constraint between the old and new policy before and after the update. Parameters will be updated based on the collected τ in time when T time-step is over. Adam optimizer is used for updating the policy and value network. ϵ is the clipping parameter, which is set to 0.2 in this chapter.

A_t is the advantage estimation given state s_t, which includes both the current and future rewards.

$$A_t = r_t + \gamma r_{t+1}\ + .. + \gamma^{T-t+1} r_{T-1} + \gamma^{T-t} r_T - V(s_t;\theta), \tag{4.12}$$

Here A_t is the difference between the accumulated reward and the estimated state value $V(s_t;\theta)$. In actor-critic algorithms, the advantage function is the difference between the accumulated reward and the estimated average reward, defined as value function $V(s_t;\theta)$.

State and Action Generally, the state comprises sufficient information from the environment for the agent to take actions. Other than directly taking the input image patches as the state like ADnet [83], in our proposed method, all the response maps produced by corresponding CF models are used as the state. In the proposed visual tracking framework, an action is defined to select one response map among all candidates by the agent. Actions are sampled from a policy distribution π, and the action with the highest score is more likely being chosen by the agent. The selected response map is used to generate tracking results in the current frame, which will be used to update CF models. These updated CF models will generate response maps in the following frames, which serve as the state of the next time slot. This above state transition process will repeat until the last frame.

Reward The reward function is defined as $r_t = g(s_t, a_t)$. A total accumulated reward can be produced until the termination time-step T. At termination time-step T, the tracker reaches the end of the sequence or it fails to locate position inside the image.

$$g(s_t, a_t) = \begin{cases} IOU + 1 & IOU > 0.7 \\ -1 & IOU < 0.2 \\ -0.1 & otherwise \end{cases}, \tag{4.13}$$

where IOU denotes the overlap ratio between tracking result and the ground-truth.

Correlation Filter Update In order to build a better discriminative appearance model, we keep k CF model in our framework, including one initial model, one accumulated model, and $k-2$ dynamic models. Siamese trackers only compare the difference between candidates in search image and the ground-truth in the first frame. So we continue to have the initial CF in our model pool without any updates. Model drift would easily happen when the tracker lost the memory of the original targets. Also, we always keep another accumulated CF model in our model pool in order to better adapt to the viewpoint change, deformation and other variations. Between these two typical situations, $k-2$ dynamic CFs play the role of 'peacemaker,' and the update for dynamic CFs only activated when chosen by the decision net. All the k models are initialized by the given tracking target, and the dynamic models are adaptively updated, during the tracking process.

The decision-making process of our approach is shown in Fig. 4.6.

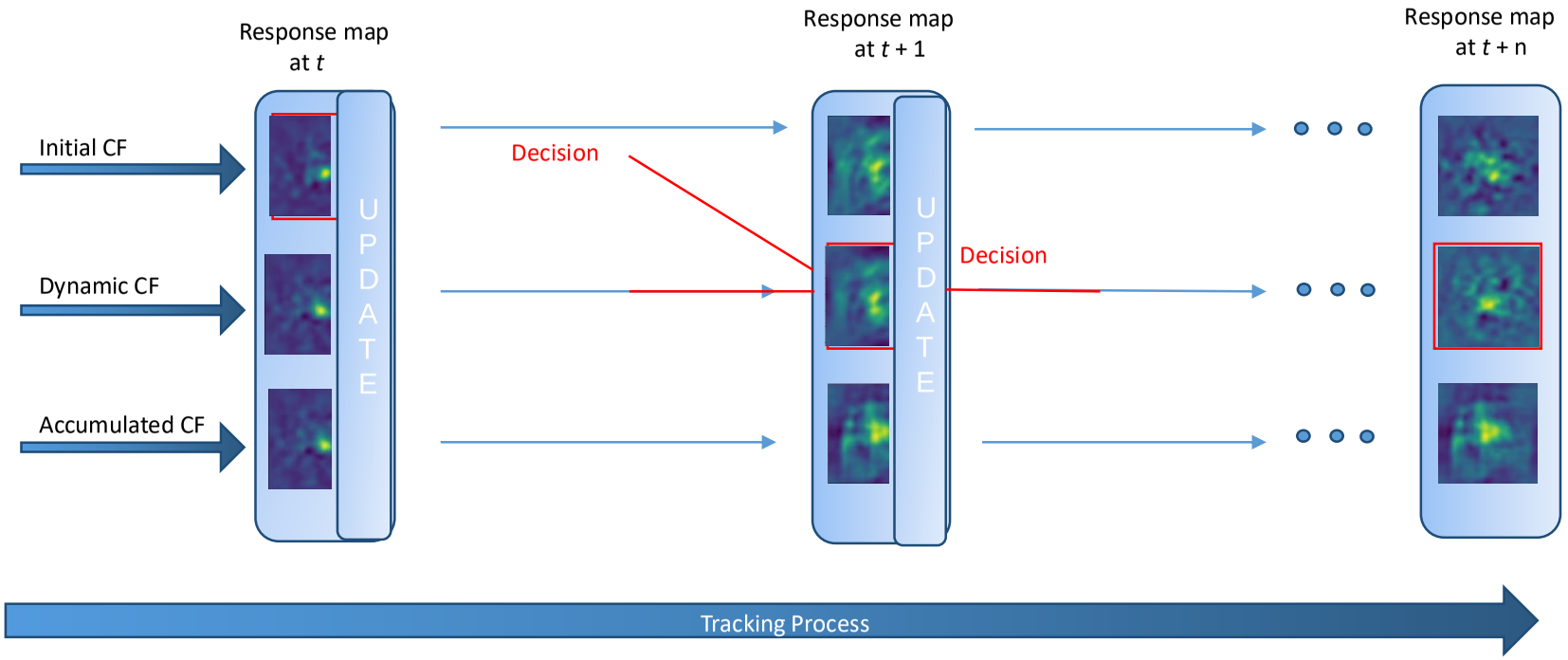


Fig. 4.6: Decision making in the tracking process. CF models numbers shown in the example is $k = 3$. As shown in this figure, an initial model is chosen at time t which result in the dynamic model unchanged until the time $t + 1$ while only the accumulated model is updated. After that, both the dynamic model and the accumulated model are updated at time $t + 1$ because of the activation of the dynamic model.

4.2.3 Decision Network

The response maps generated by the correlation filters are input to the decision network for selection. The decision network includes two branches, the policy net branch, and the value net branch which act as an actor-critic framework. As described in Table 4.1, the two branches have separate convolutional layers, one shared fully connected layer and another separate fully connected layers.

Response maps of a new input frame are resized to $64 \times 64 \times 3$ image and fed to the network as input, and here we call it the state or observation. Then the policy net branch will produce a distribution over all actions. It is worth noticing that action probability distribution is generated through beta distribution [124]. Other than the epsilon greedy policy, we use beta distribution to generate the actions during the training. It works similar to use the Gaussian policy, which can make a balance between exploration and exploitation. Finally, an action with the highest probabilities is selected.

Unlike the policy gradient algorithm for online adaptation in ADnet [83], we adopt the actor-critic framework. An expected accumulated reward is generated by the value function for one specific policy, which guides the "actor" (policy) to learn by taking feedback from the "critic" (value function) and reduces the variance of policy gradient during the training.

Table 4.1: The structure of our decision network. ($C5 \times 5 - 32S2$ means 32 filters of size 5×5 and stride 2. FC512 indicates dimension 512.)

Layers	#1	#2	#3	#4	#5
Policy net	$C5 \times 5$ $-32S2$	$C3 \times 3$ $-32S2$	$C3 \times 3$ $-32S2$	FC512	FC512
Value net	$C5 \times 5$ $-32S2$	$C3 \times 3$ $-32S2$	$C3 \times 3$ $-32S2$		FC512

4.2.4 Reinforcement Training with PPO

Environment Setup

To avoid over-fitting, we used a large-scale video detection dataset VID [125] for training our tracker. VID consists of 30 object categories, which is a subset of 200 categories in the object detection dataset. We sub-sampled the dataset and choose videos whose target size is less than 60% of their frame size.

To improve the training efficiency, we first test all selected videos via CF tracker. Based on the tracking accuracy, all the sequences are classified into three categories, including easy sequence, extremely hard sequences, and moderate sequences [126]. We exclude easy and extremely hard sequences from the training set, since (1) easy sequences will produce k similar good responses maps that vague the decision criterion, and (2) those extremely hard sequences can provide less valid samples and ambiguous labels for RL training.

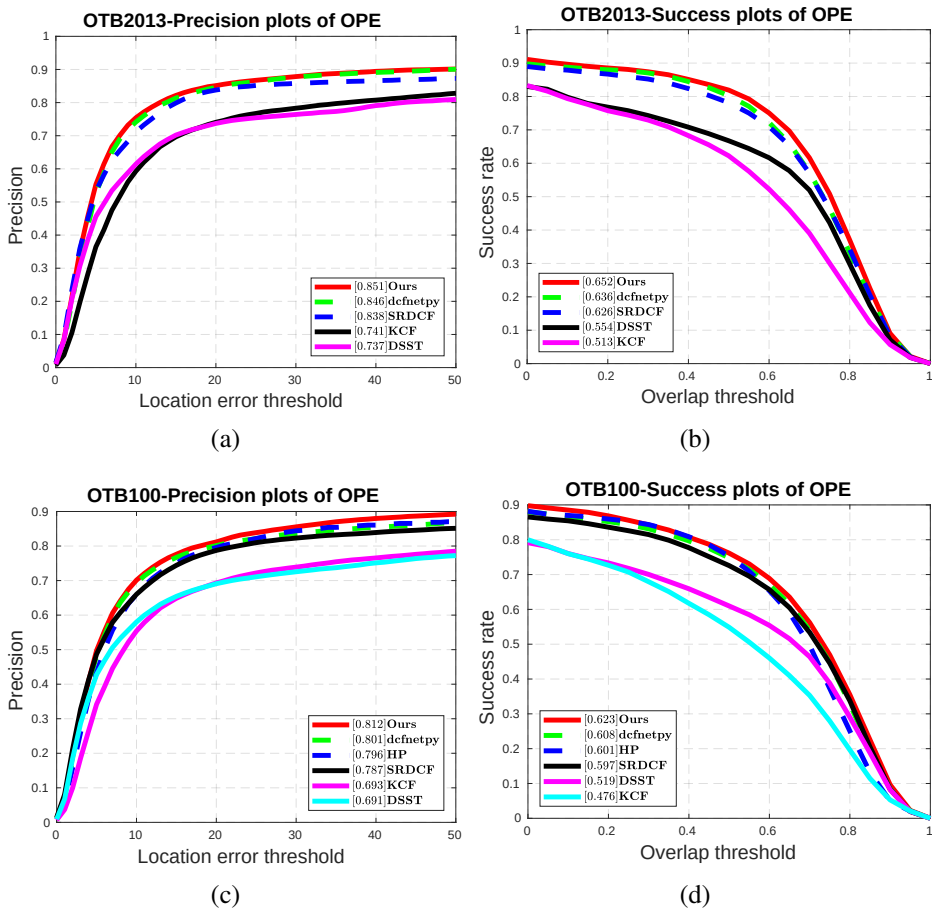
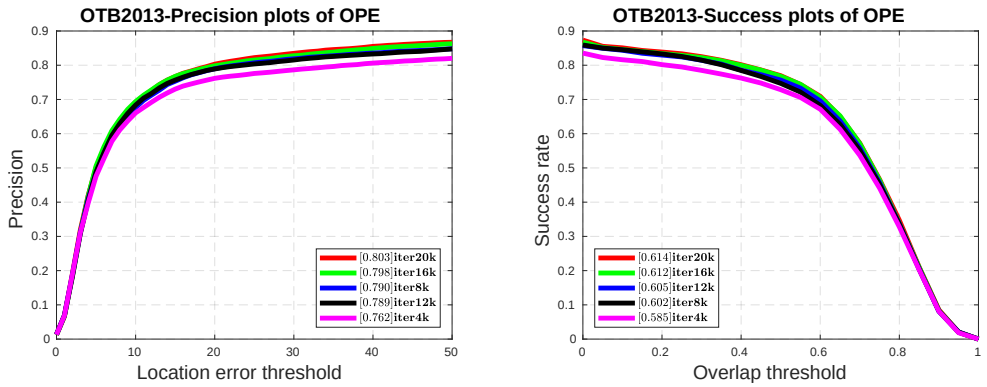


Fig. 4.7: Precision and success plots of overall performance comparison for the videos in the benchmark [3]. Average distance precision and overlap success rate are reported. Listed CF based trackers are DSST [60], KCF [46], SRDCF [62], dcfnet [67], and HP [123].

Training Process

A training batch consists of randomly sampled sub-sequences and its ground-truth from the prepared database. It is noteworthy that the unexpected tracking failure would produce a series of useless negative samples, which means the length of training sequences should be limited. While a short sequence clip usually contains insufficient, one-sided



(a) (b)

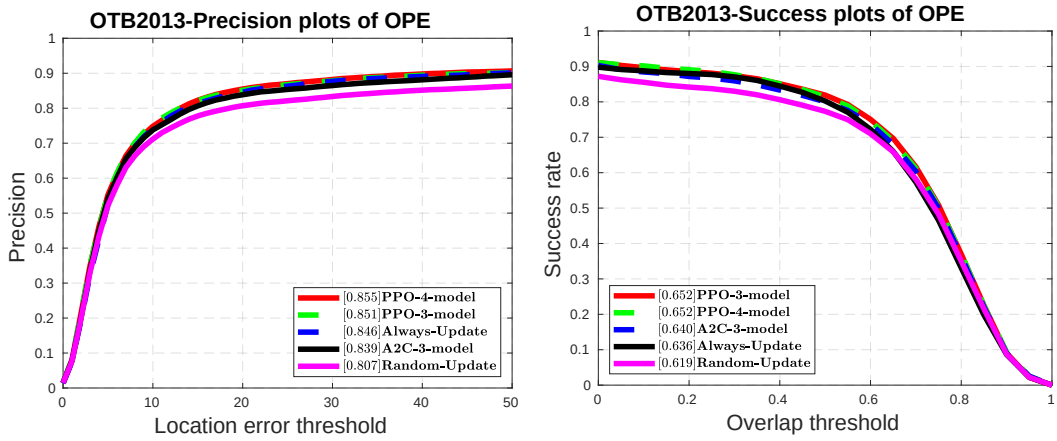
Fig. 4.8: Tracking performance comparison with various reinforcement training iterations. Five different snapshots are shown, and the OPE performance increases with the training iteration number.

information that is bad for the training. In order to boost our training process, a length of 50 is exploited in our experiments. A simulation is used to generate a series of actions by the decision net, i.e., choosing one among different response maps from stored CF models. Rewards will be obtained when the simulation is over, the right actions with high expected returns will be encouraged with high rewards. Finally, our decision net is trained to recognize appropriate CF models by optimizing the clipped surrogate objective function (4.10).

Generally, in each iteration, our on-policy RL algorithm updates θ several times by gradient ascent, i.e., $\theta \leftarrow \theta + \Delta\,\theta$. If the new policy π or the new state value V changes exceed a certain threshold, the clipped function will limit the network parameter update, which effectively constrains the variation caused by a challenging tracking sequence. This mechanism improves training stability.

4.3 Experiments

In this section, we detail our experimental setup and the parameters we used during the training and testing. Quantitative and qualitative experiments have been conducted on popular visual tracking benchmark datasets, namely the OTB2013 and OTB100. We compared our proposed algorithm with the other five CF-based tracking frameworks. Meanwhile, we validated the effectiveness of our proposed method by conducting various ablation studies. For fair comparisons, no additional modification is allowed during the evaluation. The experiments were conducted on a system with an E5-2620v3 2.4GHz CPU having 32 GB memory and a GTX TITAN X GPU using MATLAB2017b and PyTorch.



(a) (b)

Fig. 4.9: Tracking performance comparison of three different model update strategies: always updating CF model, random updating and the proposed updating by decision(PPO-3-model, PPO-4-model, A2C-3-model).

4.3.1 Experimental Setup

In the tracking process, the searching image within the current frame is twice the target size on the horizontal and vertical directions. In order to cover different scale changes, three scaled versions of the search image are used to find the best scale that fits the scale change. The scale parameter is set to 1.025. If not explicitly specified, three CF models are maintained in our experiments, i.e., $k = 3$, including the initial model for tracking, the dynamic model, and the accumulated model. The accumulated model is updated at each frame, while the initial model is kept unchanged. The dynamic model is updated once it is selected by the decision network. For the dynamic model and accumulated model, the average moving parameter 0.05 is used, and new CF models will replace old models.

4.3.2 Quantitative and Qualitative Comparisons on Benchmarks

We evaluated our method in comparison with existing CF trackers on the popular visual tracking benchmarks, Object Tracking Benchmark (OTB) [3]. Tracking algorithms KCF [46], DSST [60], SRDCF [62], DCFnet [67] and HP [123] are evaluated for comparison. The *dcfnetpy* is our implemented algorithm of DCFnet [67] in python, which achieved similar performance as reported in [67]. Two standard evaluation metrics, namely distance precision (DP) and overlap success (OS) rate are used to evaluate trackers' performance. DP is the frame proportion of the predicted position within a given threshold. The overlap success rate is defined as the percentage of frames that overlap between predicted location and ground-truth surpassing the threshold.

Algorithm 2: RL training via PPO

Input:
Random sampled tracking sequence of length L, along with it ground-truth G
Decision network $D(\theta)$
Output: Updated Decision network D

Initialize CF model M_0 with the ground-truth
Set history CF model M_1, .. , M_{K-1} = M_0
for $t = 1$ *to* L **do**
  for $i = 1$ *to K* **do**
    Calculate response maps P_i with each M_i
    Produce confidence score via decision net $\pi(a_t|s_t;\theta)$ for each P_i
  end
  Choose the prediction map with the maximum confidence;
  Localize the target according to chosen P_m;
  Obtain reward r_m
  Update corresponding CF models and save to history;
end
Sum discounted reward as return
Update Decision network $D(\theta)$ by Adam with equation (4.10) for $d = 10$ times

Table 4.2: A comparison of our approach with other CF-based trackers. The mean overlap precision (OS) (%) and distance precision (DP) (%) over all the videos in the OTB2013 dataset are presented. DP at a threshold of 20 pixels, overlap success (OS) rate at an overlap threshold 0.6.

Method	Proposed	*dcfnetpy*	SRDCF [62]	DSST [60]	KCF [46]
OS (%)	**74.58**	72.23	70.98	61.65	52.25
DP (%)	**85.12**	84.59	83.79	73.70	74.06

Table 4.3: A comparison of our approach with other CF-based trackers. The mean overlap precision (OS) (%) and distance precision (DP) (%) over all the 100 videos in the OTB100 dataset are presented. DP at a threshold of 20 pixels, overlap success (OS) rate at an overlap threshold 0.6.

Method	Proposed	*dcfnetpy*	SRDCF [62]	DSST [60]	KCF [46]
OS (%)	**68.89**	67.07	65.67	55.39	46.03
DP (%)	**81.19**	80.13	78.74	69.10	69.31

Quantitative Comparison Overall performance comparison for the 51 videos in the benchmark [3] is reported in Fig. 4.7, which includes both precision and success plots. It can be observed that in success plots, our proposed algorithm is always above other

Table 4.4: Tracking performance comparison with various reinforcement training iterations. On OTB2013, DP at a threshold of 20 pixels, overlap success (OS) rate at an overlap threshold 0.6.

Iteration	*4k*	*8k*	*12k*	*16k*	*20k*
OS (%)	67.2	68.3	69.6	70.6	70.9
DP (%)	76.2	78.9	79.0	79.8	80.3

Table 4.5: Tracking performance comparison of 5 different model update strategies and test On OTB2013, DP at a threshold of 20 pixels, overlap success (OS) rate at an overlap threshold 0.6.

Method	PPO-4-model	PPO-3-model	A2C-3-model	Always-Update	Random-Update
OS (%)	**75.05**	**74.58**	73.74	72.33	71.00
DP (%)	**85.48**	**85.12**	83.85	84.59	80.72

trackers for overlap threshold above 0.5. The performance gain is increasing with overlap threshold, showing our proposed method consistently contributes to the tracking accuracy with various overlap threshold.

Table 4.2 is comparisons of our approach with other CF-based trackers. The mean overlap precision (OS) and distance precision (DP) over all the OTB datasets are presented. The results are obtained with DP at a threshold of 20 pixels, overlap success(OS) rate at an overlap threshold of 0.6. Results show that our algorithm performs favorably against other CF methods for a common setting. Among the existing CF trackers, our proposed method achieves the best results with an OS of 68.89%, DP of 81.19% on OTB100. Our achieved OS and DP are respectively 1.82% and 1.06% higher than that of CF models without model selection (*dcfnetpy*).

In Fig. 4.10, the performance of 5 CF-based trackers for 11 attributes on OTB100 is reported, including background clutter, low resolution, scale variation, illumination variation, deformation, motion blur, in-plane rotation, occlusion, and out-of-view. Generally, our proposed tracker achieves superior accuracy compared to other CF trackers for most of the attributes. Due to the multiple model selection, our method is able to handle occlusion better during the tracking, and the results in 48 occlusion sequences improve by 3.7% in success rate and 3.1% in precision compared with the always updating strategy (*dcfnetpy*). Similarly, our proposed method works well in 14 out of view sequences and in 9 low-resolution sequences.

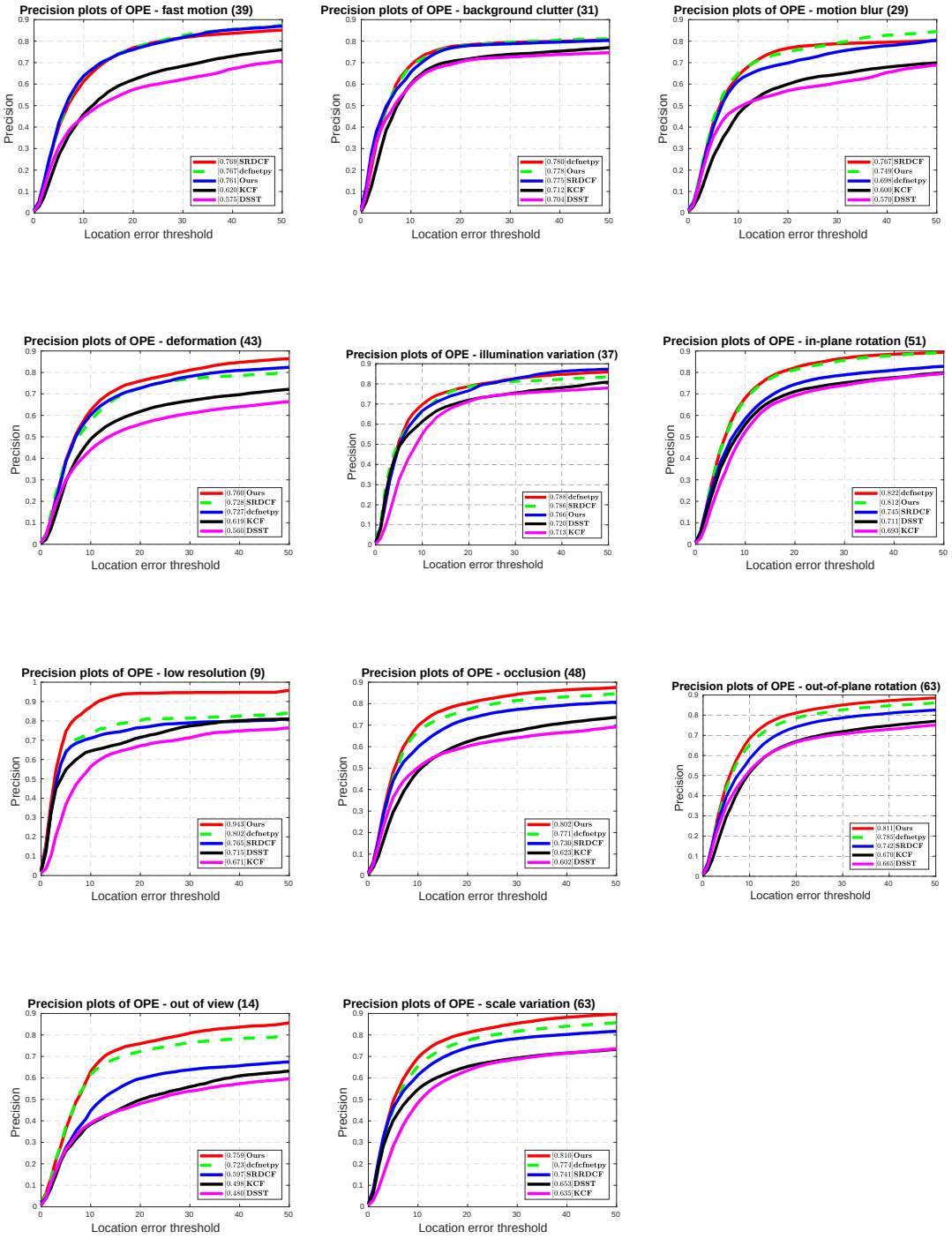


Fig. 4.10: The performance of 5 CF-based trackers for 11 attributes on OTB100, which contains 100 video sequences. *dcfnetpy* is our python implementation of DCFNET. Our proposed model achieves higher success rate and precision compared with others.

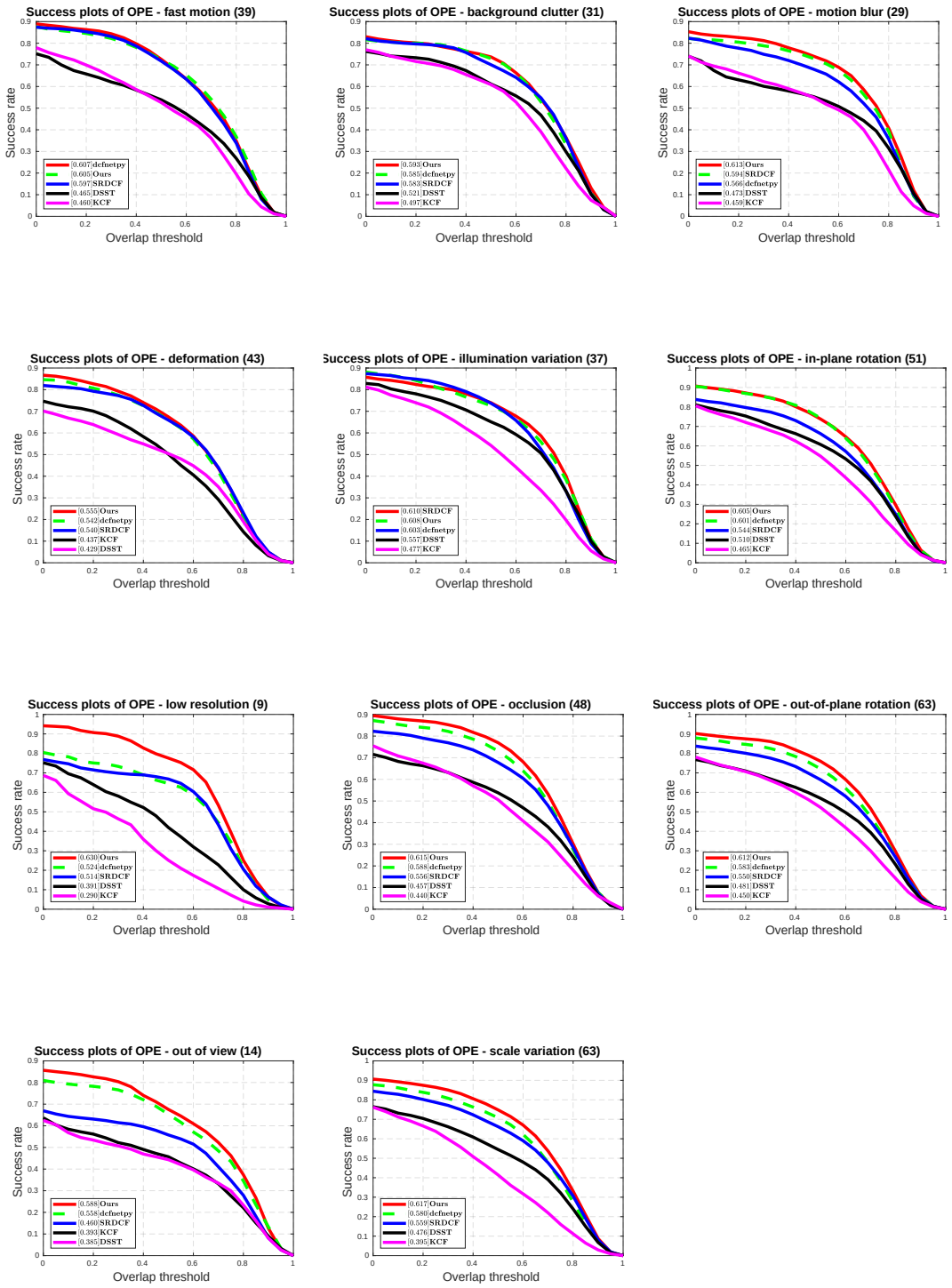


Fig. 4.11: The performance of 5 CF-based trackers for 11 attributes on OTB100, which contains 100 video sequences.

Qualitative Comparison Fig. 4.12 presents the superiority of our algorithm qualitatively compared to the other 4 CF trackers on seven challenging sequences. The CF2, DSST methods lose track of the target gradually due to significant occlusion and motion blur in Box and Girl sequences. The SRDCF, KCF, CF2 trackers are not able to keep tracking the target after occlusion and illumination changes in Box and Girl2 sequences. It can also be observed that when scale variation and occlusion happen as in Dragonbaby, the DSST and KCF trackers do not perform well. Other trackers fail in the presence of out-of-plane rotation, scale variation, and fast motion. It is noticed that our proposed multiple model selection could discover the missing target after a long-term tracking failure, while other trackers can hardly recover from the drifting. Overall, our proposed tracker is able to alleviate the drifting issue in many challenging sequences.

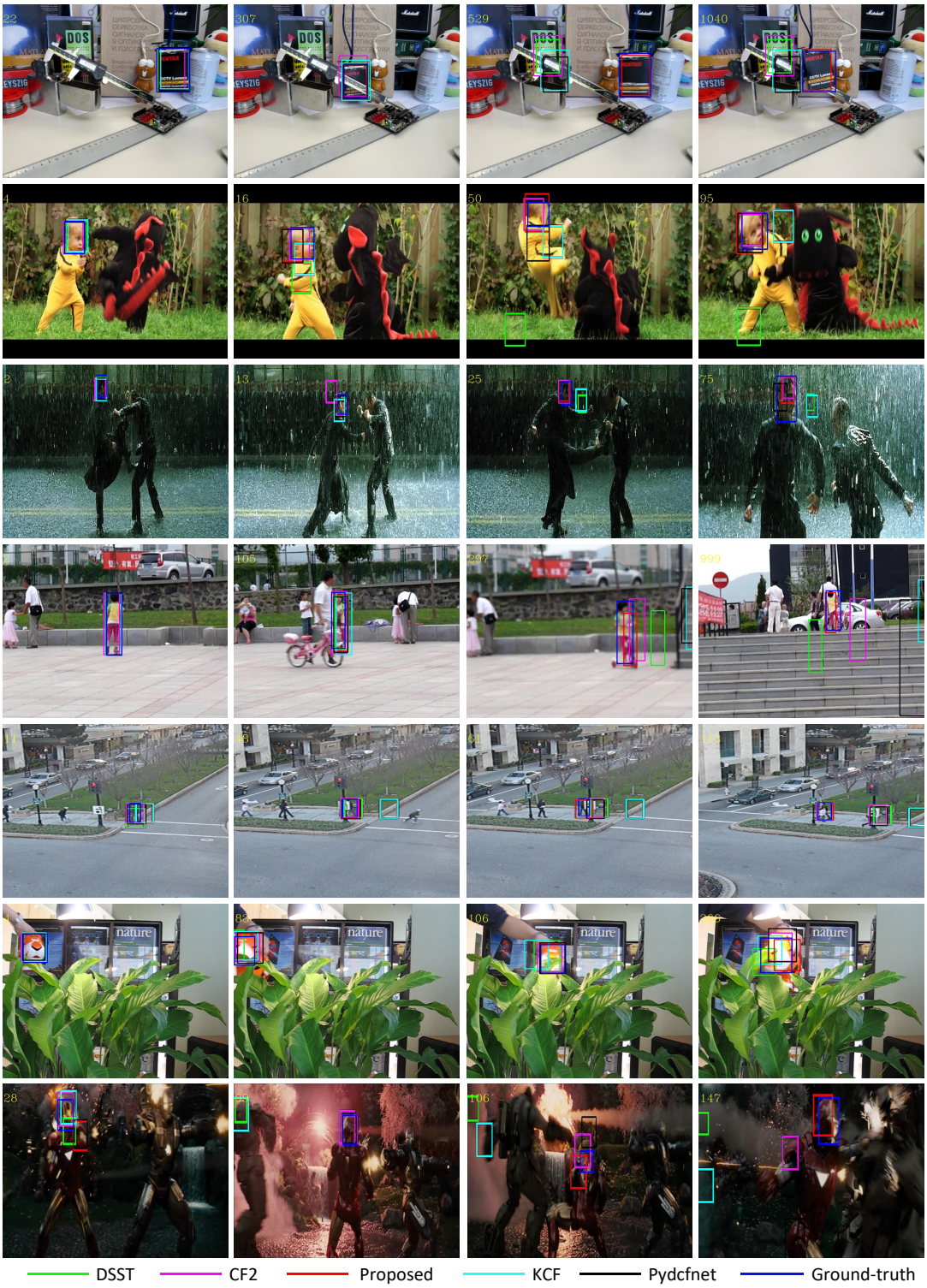


Fig. 4.12: Visualizations of our tracking results(Box, DragonBaby, Matrix, Girl2, Human, Tiger, Ironman). Green, Purple, Red, Light Blue, and Black box denote tracking results of DSST, CF2, Proposed, KCF, *pydcfnet*, respectively. Blue box is the ground-truth box, Yellow numbers on the top-left corners indicate frame numbers.

4.3.3 Ablation Study

We conducted some ablation studies to demonstrate the effectiveness of our method. In Fig. 4.8, performance is reported with different training iterations on OTB2013. The precision and success rates increase with the iteration, proving that the reinforcement learning process effectively guides the optimization.

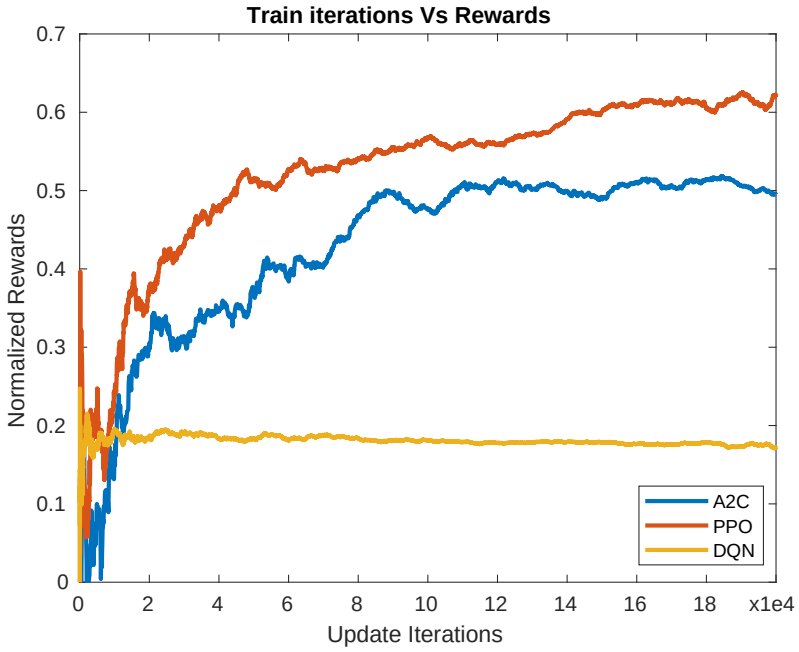


Fig. 4.13: Normalized Rewards vs Iteration Number through train process, A2C [81], PPO [69], and DQN [86]

We also conducted additional experiments with different CF model selection/updating schemes. The ”always-update” scheme always uses the latest CF model to find the tracking object and updates the model at each frame. The ”random-update” scheme randomly select a model among the initial model, dynamic model, and accumulated model, and updates the randomly selected model at each frame. We set a dynamic model with number 1 and 2, respectively, in our experiments, which are denoted by PPO-3-model and PPO-4-model. The performance comparison results are plotted in Fig. 4.9, and numeric results are reported in Table 4.4. Overall, our model selection scheme outperforms both the ”always-update” and ”random-update” schemes, showing that by using the proposed CF selection strategy, our decision network is able to choose the most suitable CF for visual tracking, and to a certain extent, the model drift has been reduced. PPO-model-3 and PPO-model-4 lead to similar performance for both metric OS and DP. Therefore, we use

PPO-model-3 model throughout the chapter for performance evaluation due to its lower complexity.

We carried out a one-step supervised learning experiment for the CF model selection. We collected tracking response maps by running our CF tracker on visual object tracking dataset for each frame, the responses maps which bring the highest IOU are recorded as positive, while others as negative for classification model training. For this purpose, we randomly sampled the training data prior to the running of the experiments and consolidated them into a dedicated sample pool. The ratio of 1:2 is maintained between the positive and negative samples, respectively. We kept adopting a similar model structure to the approach used during RL training, with the last layer outputting a binary classification probability for CF model selection. We also used standard Cross-Entropy loss during the training process. Our observation was that we failed to make the network converge for the given dataset in the supervised learning experiment.

We also employ different reinforcement learning algorithms to replace the proximal policy optimization. First, we disable the Clipped Surrogate term and degrade it into a basic synchronous advantage actor-critic model (A2C [81]). The policy/value network structures and parameters are kept the same. The update is performed after the 4 actor-learners finishing collecting data, in order to improve the training stability. Moreover, we further degrade the RL algorithm into a DQN [86]. The agent's experiences state at each timestep is stored to perform experience replay, Q-learning updates are applied by random sampling from the experience pool. The train reward with update iterations is shown in Fig. 4.13. It shows that an improvement of the reward due to the advantage actor-critic algorithm A2C and PPO, while the traditional DQN does not work for the model selection under a similar training setting. It is also important to note that the A2C algorithm takes 50% more time to reach the same update iteration number of PPO in our experiments. Also, the PPO algorithm ends up with higher rewards than the A2C algorithm, and better tracking performance is achieved, as reported in Table 4.5.

4.4 Conclusions

In this chapter, we have proposed a novel approach for CF-based visual tracking. In our approach, multiple CF models are updated and maintained in parallel, and an optimal model is selected on demand using deep reinforcement learning. The proposed algorithm learns the model selection policy with the proximal policy optimization algorithm, while utilizing the selected CF model to conduct object tracking.

We show that the model selection via response map can effectively overcome the model drifting issues, and enhance the robustness of the trackers. Our exhaustive experi-

mental evaluation using two key benchmarks, covering both the quantitative and qualitative aspects, show that our approach can handle a number of tracking challenges and can offer substantially better tracking performance when compared to traditional CF-based trackers.

Chapter 5

Conclusions

In this chapter, the final summary of this thesis will be presented, followed by some future works in relevant research directions.

5.1 Summary

The person re-identification and tracking problems are important in video surveillance systems. Among the tasks, the feature learning and model update are challenging and essential problems. A more discriminative feature representation could lead to an improvement in the accuracy, and a better model update strategy could increase the robustness and prevent a model drift. Base on this, the first purpose of the thesis was to investigate the features in Person re-identification tasks. To make this task, we proposed a novel Individual Aggregation Network that can not only accurately localize pedestrians but also minimize feature representations of intra-person variations. In particular, we built the network upon the state-of-the-art object detection framework Faster R-CNN, so that high-quality region proposals for pedestrians can be produced in an online manner for person search. In addition, to relieve the negative effect caused by various visual appearances of the same individual, we introduced the center loss to increase the intra-class compactness of feature representations. The center loss encourages learned pedestrian representations from the same class to share similar feature characteristics. Meanwhile, we also performed a neural network compatibility study for center loss, and we explained why dropout is not compatible with center loss. We study this phenomenon in both analytic and experimental ways. Finally, extensive experiments were performed on two benchmarks, i.e., CUHK-SYSU and PRW, show that IAN achieves the state-of-the-art performance on both datasets, and well demonstrate the superiority of the proposed network. Also, our proposed method can be embedded in any CNN-based person search framework for improving performance.

In this thesis, we also propose an ensemble siamese tracker, to handle the update problem in visual object tracking. We considered to merge and update the features with

tracking results in recent frames instead of solely considering the first frame. Specifically, the tracking results in 25 recent frames are used to adjust the model for a continuous target change. Meanwhile, we combine adaptive candidate sampling strategy and large displacement optical flow method with our method to further improve the performance.

Finally, to address the challenging task of model update problems in correlation filter based methods, we proposed a reinforcement learning-based approach for optimal model selection. The decision network could select an optimal model among multiple CF models that are updated and maintained in parallel. This approach addresses a number of concerns that arise from a single CF model, such as drift. This is the first time that reinforcement learning is utilized for model selection among multiple CF models. We utilized a lightweight feature extractor and proposed a small decision network so that the proposed approach can be deployed in realtime applications, where the frame rates are high.

Overall, in this thesis, we mainly study the two vital tasks in video surveillance systems: person re-identification and object tracking. We propose novel approaches to improve the abilities of feature representation and enhance the model/feature update strategy.

5.2 Futureworks

At the time of concluding this manuscript, several exciting perspectives can be proposed to further continue the work done in this thesis.

The primary points concern the use of the center loss. Because center loss needs to track the feature centers of all classes, one limitation of the proposed method is its large GPU memory requirement. It can be further investigated to find a more efficient way to represent the feature center. Furthermore, interesting future research for this work might be to use better supervised and more clean labels such as 'pixel-level' annotations instead of rectangle boxes that contain noisy background information. Besides, the dataset in this research is limited, and it is worth studying the unsupervised/semi-supervised way to train the network with more data.

Moreover, in the ensemble siamese tracker work, a memory mechanism could be further developed, by which the matching unit could benefit from the long term storage. We think that by using long-term temporal information from the video sequences could result in better accuracy and handle more difficult tracking situations. Besides our proposed update method, we consider to design an update network to enable the network itself to learn to fuse the accumulated templates for the tracking target, and we could decrease the hyperparameters through the learning process. Overall, the updating problem in siamese based tracking algorithms is imperative, and many pieces of research could be investigated on this problem.

Finally, in the RL based model selection work, we apply the reinforcement learning on the model updating problem. We use a simple reward function, which only considers the success/failure of the tracking process, which is simple while not efficient. We believe that a better reward function could be designed, and it should reflect more about the update itself instead of barely the tracking performance. On the other direction, when obtaining discriminative features for tracking, large models tend to introduce severe over-fitting problems. So it is necessary to study the network structure and provide an efficient feature network for the kinds of these trackers. It is also interesting to define the tracking problem as a continuous control problem, in which the whole task could be learned and benefited through the large scale reinforcement learning.

Appendix: A list of Publications

Here is a brief list of my research publications during my Ph.D. studies:

1. Yanchun Xie, Jimin Xiao, Mingjie Sun, Chao Yao, Kaizhu Huang: Matching Feature Matters: End-to-End Learning for Neural Texture Transfer (submitted to ECCV2020)

2. Yanchun Xie, Jimin Xiao, Kaizhu Huang, Jeyarajan Thiyagalingam, Yao Zhao: Correlation Filter Selection for Visual Tracking Using Reinforcement Learning. IEEE Transactions on Circuits and Systems for Video Technology (TCSVT). 30(1): 192-204 (2020)

3. Jimin Xiao, Yanchun Xie, Tammam Tillo, Kaizhu Huang, Yunchao Wei, Jiashi Feng: IAN: The Individual Aggregation Network for Person Search. Pattern Recognition (PR). 87: 332-340 (2019)

4. Chenru Jiang, Jimin Xiao, Yanchun Xie, Tammam Tillo, Kaizhu Huang: Siamese network ensemble for visual tracking. Neurocomputing 275: 2892-2903 (2018)

5. Yanchun Xie, Jimin Xiao, Tammam Tillo, Yunchao Wei, Yao Zhao: 3D video super-resolution using fully convolutional neural networks. IEEE International Conference on Multimedia & Expo(ICME): 1-6 (2016)

Reference

[1] Tong Xiao, Shuang Li, Bochao Wang, Liang Lin, and Xiaogang Wang. End-to-end deep learning for person search. *arXiv preprint arXiv:1604.01850*, 2016.

[2] Tong Xiao, Shuang Li, Bochao Wang, Liang Lin, and Xiaogang Wang. Joint detection and identification feature learning for person search. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 3376–3385, 2017.

[3] Yi Wu, Jongwoo Lim, and Ming-Hsuan Yang. Online object tracking: A benchmark. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 2411–2418, 2013.

[4] Ran Tao, Efstratios Gavves, and Arnold W M Smeulders. Siamese instance search for tracking. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1420–1429, 2016.

[5] Apurva Bedagkar-Gala and Shishir K Shah. A survey of approaches and trends in person re-identification. *Image and Vision Computing*, 32(4):270–286, 2014.

[6] Shaogang Gong and Tao Xiang. Person re-identification. In *Visual Analysis of Behaviour*, pages 301–313. 2011.

[7] Xiaokai Liu, Hongyu Wang, Jie Wang, and Xiaorui Ma. Person re-identification by multiple instance metric learning with impostor rejection. *Pattern Recognition*, 67(C):287–298, 2017.

[8] Xiaolong Ma, Xiatian Zhu, Shaogang Gong, Xudong Xie, Jianming Hu, Kin Man Lam, and Yisheng Zhong. Person re-identification by unsupervised video matching. *Pattern Recognition*, 65(C):197–210, 2016.

[9] Shengyong Ding, Liang Lin, Guangrun Wang, and Hongyang Chao. Deep feature learning with relative distance comparison for person re-identification. *Pattern Recognition*, 48(10):2993–3003, 2015.

[10] Lin Wu, Yang Wang, Junbin Gao, and Xue Li. Deep adaptive feature embedding with local sample distributions for person re-identification. *Pattern Recognit.*, 73:275–288, 2018.

[11] Zhicheng Zhao, Binlin Zhao, and Fei Su. Person re-identification via integrating patch-based metric learning and local salience learning. *Pattern Recognit.*, 75:90–98, 2018.

[12] Yuanlu Xu, Bingpeng Ma, Rui Huang, and Liang Lin. Person search in a scene by jointly modeling people commonness and person uniqueness. In *International Conference on Multimedia*, pages 937–940, 2014.

[13] Liang Zheng, Hengheng Zhang, Shaoyan Sun, Manmohan Chandraker, Yi Yang, Qi Tian, et al. Person re-identification in the wild. In *IEEE Conference on Computer Vision and Pattern Recognition*, volume 1, page 2, 2017.

[14] Yi Sun, Xiaogang Wang, and Xiaoou Tang. Deep learning face representation from predicting 10,000 classes. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1891–1898, 2014.

[15] Yi Sun, Yuheng Chen, Xiaogang Wang, and Xiaoou Tang. Deep learning face representation by joint identification-verification. In *Advances in Neural Information Processing Systems*, pages 1988–1996, 2014.

[16] Ran Tao, Efstratios Gavves, and Arnold WM Smeulders. Siamese instance search for tracking. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1420–1429, 2016.

[17] Dong Yi, Zhen Lei, Shengcai Liao, and Stan Z Li. Deep metric learning for person re-identification. In *IEEE Conference on International Conference and Pattern Recognition*, pages 34–39, 2014.

[18] Wei Li, Rui Zhao, Tong Xiao, and Xiaogang Wang. Deepreid: Deep filter pairing neural network for person re-identification. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 152–159, 2014.

[19] Ejaz Ahmed, Michael Jones, and Tim K Marks. An improved deep learning architecture for person re-identification. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 3908–3916, 2015.

[20] Douglas Gray and Hai Tao. Viewpoint invariant pedestrian recognition with an ensemble of localized features. In *European Conference on Computer Vision*, pages 262–275, 2008.

[21] Wei Li, Rui Zhao, and Xiaogang Wang. Human reidentification with transferred metric learning. In *Asian Conference on Computer Vision*, pages 31–44, 2012.

[22] Rahul Rama Varior, Bing Shuai, Jiwen Lu, Dong Xu, and Gang Wang. A siamese long short-term memory architecture for human re-identification. In *European Conference on Computer Vision*, pages 135–153, 2016.

[23] Hao Liu, Jiashi Feng, Meibin Qi, Jianguo Jiang, and Shuicheng Yan. End-to-end comparative attention networks for person re-identification. *IEEE Transactions on Image Processing*, 26(7):3492–3506, 2017.

[24] Liang Zheng, Zhi Bie, Yifan Sun, Jingdong Wang, Chi Su, Shengjin Wang, and Qi Tian. Mars: A video benchmark for large-scale person re-identification. In *European Conference on Computer Vision*, pages 868–884, 2016.

[25] Wei-Shi Zheng, Shaogang Gong, and Tao Xiang. Person re-identification by probabilistic relative distance comparison. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 649–656, 2011.

[26] Wei-Shi Zheng, Shaogang Gong, and Tao Xiang. Reidentification by relative distance comparison. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(3):653–668, 2013.

[27] Liang Zheng, Liyue Shen, Lu Tian, Shengjin Wang, Jingdong Wang, and Qi Tian. Scalable person re-identification: A benchmark. In *IEEE International Conference on Computer Vision*, pages 1116–1124, 2015.

[28] Dapeng Tao, Lianwen Jin, Yongfei Wang, and Xuelong Li. Person reidentification by minimum classification error-based kiss metric learning. *IEEE Transactions on Cybernetics*, 45(2):242–252, 2015.

[29] Liliang Zhang, Liang Lin, Xiaodan Liang, and Kaiming He. Is faster r-cnn doing well for pedestrian detection? In *European Conference on Computer Vision*, pages 443–457, 2016.

[30] Shanshan Zhang, Rodrigo Benenson, and Bernt Schiele. Filtered channel features for pedestrian detection. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1751–1760, 2015.

[31] Jianan Li, Xiaodan Liang, ShengMei Shen, Tingfa Xu, Jiashi Feng, and Shuicheng Yan. Scale-aware fast r-cnn for pedestrian detection. *IEEE Transactions on Multimedia*, 20(4):985–996, 2017.

[32] Yandong Wen, Kaipeng Zhang, Zhifeng Li, and Yu Qiao. A discriminative feature learning approach for deep face recognition. In *European Conference on Computer Vision*, pages 499–515, 2016.

[33] Amit Adam, Ehud Rivlin, and Ilan Shimshoni. Robust fragments-based tracking using the integral histogram. In *IEEE Conference on Computer Vision and Pattern Recognition*, volume 1, pages 798–805, 2006.

[34] Junseok Kwon and Kyoung Mu Lee. Tracking by sampling and integratingmultiple trackers. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36(7):1428–1441, 2013.

[35] Abhishek Kumar Chauhan and Prashant Krishan. Moving object tracking using gaussian mixture model and optical flow. *International Journal of Advanced Research in Computer Science and Software Engineering*, 3(4), 2013.

[36] David A Ross, Jongwoo Lim, Ruei-Sung Lin, and Ming-Hsuan Yang. Incremental learning for robust visual tracking. *International Journal of Computer Vision*, 77(1-3):125–141, 2008.

[37] Min Tian, Weiwei Zhang, and Fuqiang Liu. On-line ensemble svm for robust object tracking. In *Asian Conference on Computer Vision*, pages 355–364, 2007.

[38] Kaihua Zhang, Lei Zhang, and Ming-Hsuan Yang. Fast compressive tracking. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36(10):2002–2015, 2014.

[39] Boris Babenko, Ming-Hsuan Yang, and Serge Belongie. Robust object tracking with online multiple instance learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(8):1619–1632, 2010.

[40] Nan Jiang, Wenyu Liu, and Ying Wu. Learning adaptive metric for robust visual tracking. *IEEE Transactions on Image Processing*, 20(8):2288–2300, 2011.

[41] Hyeonseob Nam and Bohyung Han. Learning multi-domain convolutional neural networks for visual tracking. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4293–4302, 2016.

[42] Naiyan Wang, Siyi Li, Abhinav Gupta, and Dit-Yan Yeung. Transferring rich feature hierarchies for robust visual tracking. *arXiv preprint arXiv:1501.04587*, 2015.

[43] H Fan and Ling H SANet. Structure-aware network for visual tracking. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops CVPRW*, pages 2217–2224, 2016.

[44] João F Henriques, Rui Caseiro, Pedro Martins, and Jorge Batista. Exploiting the circulant structure of tracking-by-detection with kernels. In *European Conference on Computer Vision*, pages 702–715, 2012.

[45] Naiyan Wang and Dit-Yan Yeung. Learning a deep compact image representation for visual tracking. In *Advances in Neural Information Processing Systems*, pages 809–817, 2013.

[46] João F Henriques, Rui Caseiro, Pedro Martins, and Jorge Batista. High-speed tracking with kernelized correlation filters. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 37(3):583–596, 2015.

[47] Lijun Wang, Wanli Ouyang, Xiaogang Wang, and Huchuan Lu. Visual tracking with fully convolutional networks. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3119–3127, 2015.

[48] Luca Bertinetto, Jack Valmadre, Joao F Henriques, Andrea Vedaldi, and Philip HS Torr. Fully-convolutional siamese networks for object tracking. In *European Conference on Computer Vision*, pages 850–865, 2016.

[49] Martin Danelljan, Andreas Robinson, Fahad Shahbaz Khan, and Michael Felsberg. Beyond correlation filters: Learning continuous convolution operators for visual tracking. In *European Conference on Computer Vision*, pages 472–488, 2016.

[50] Martin Danelljan, Goutam Bhat, F Shahbaz Khan, and Michael Felsberg. Eco: Efficient convolution operators for tracking. In *Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA*, pages 21–26, 2017.

[51] Bo Li, Junjie Yan, Wei Wu, Zheng Zhu, and Xiaolin Hu. High performance visual tracking with siamese region proposal network. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8971–8980, 2018.

[52] Yibing Song, Chao Ma, Xiaohe Wu, Lijun Gong, Linchao Bao, Wangmeng Zuo, Chunhua Shen, Rynson WH Lau, and Ming-Hsuan Yang. Vital: Visual tracking via adversarial learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8990–8999, 2018.

[53] Martin Danelljan, Goutam Bhat, Fahad Shahbaz Khan, and Michael Felsberg. Atom: Accurate tracking by overlap maximization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4660–4669, 2019.

[54] Peixia Li, Boyu Chen, Wanli Ouyang, Dong Wang, Xiaoyun Yang, and Huchuan Lu. Gradnet: Gradient-guided network for visual object tracking. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 6162–6171, 2019.

[55] Esteban Real, Jonathon Shlens, Stefano Mazzocchi, Xin Pan, and Vincent Vanhoucke. Youtube-boundingboxes: A large high-precision human-annotated data set for object detection in video. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5296–5305, 2017.

[56] Heng Fan, Liting Lin, Fan Yang, Peng Chu, Ge Deng, Sijia Yu, Hexin Bai, Yong Xu, Chunyuan Liao, and Haibin Ling. Lasot: A high-quality benchmark for large-scale single object tracking. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5374–5383, 2019.

[57] Zhipeng Zhang and Houwen Peng. Deeper and wider siamese networks for real-time visual tracking. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4591–4600, 2019.

[58] Qiang Wang, Li Zhang, Luca Bertinetto, Weiming Hu, and Philip HS Torr. Fast online object tracking and segmentation: A unifying approach. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1328–1338, 2019.

[59] Paul Voigtlaender, Jonathon Luiten, Philip HS Torr, and Bastian Leibe. Siam r-cnn: Visual tracking by re-detection. *arXiv preprint arXiv:1911.12836*, 2019.

[60] Martin Danelljan, Gustav Häger, Fahad Khan, and Michael Felsberg. Accurate scale estimation for robust visual tracking. In *British Machine Vision Conference, Nottingham, September 1-5, 2014*, 2014.

[61] Martin Danelljan, Gustav Häger, Fahad Shahbaz Khan, and Michael Felsberg. Discriminative scale space tracking. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(8):1561–1575, 2017.

[62] Martin Danelljan, Gustav Hager, Fahad Shahbaz Khan, and Michael Felsberg. Learning spatially regularized correlation filters for visual tracking. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 4310–4318, 2015.

[63] Matthias Mueller, Neil Smith, and Bernard Ghanem. Context-aware correlation filter tracking. In *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.(CVPR)*, pages 1396–1404, 2017.

[64] Chao Ma, Jia-Bin Huang, Xiaokang Yang, and Ming-Hsuan Yang. Hierarchical convolutional features for visual tracking. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3074–3082, 2015.

[65] Martin Danelljan, Gustav Hager, Fahad Shahbaz Khan, and Michael Felsberg. Convolutional features for correlation filter based visual tracking. In *Proceedings of the IEEE International Conference on Computer Vision Workshops*, pages 58–66, 2015.

[66] Yibing Song, Chao Ma, Lijun Gong, Jiawei Zhang, Rynson WH Lau, and Ming-Hsuan Yang. Crest: Convolutional residual learning for visual tracking. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2574–2583, 2017.

[67] Qiang Wang, Jin Gao, Junliang Xing, Mengdan Zhang, and Weiming Hu. Dcfnet: Discriminant correlation filters network for visual tracking. *arXiv preprint arXiv:1704.04057*, 2017.

[68] John Schulman, Sergey Levine, Pieter Abbeel, Michael Jordan, and Philipp Moritz. Trust region policy optimization. In *International Conference on Machine Learning*, pages 1889–1897, 2015.

[69] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

[70] H Kiani Galoogahi, Ashton Fagg, and Simon Lucey. Learning background-aware correlation filters for visual tracking. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 21–26, 2017.

[71] Mengmeng Wang, Yong Liu, and Zeyi Huang. Large margin object tracking with circulant feature maps. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA*, pages 21–26, 2017.

[72] Sergey Levine, Chelsea Finn, Trevor Darrell, and Pieter Abbeel. End-to-end training of deep visuomotor policies. *The Journal of Machine Learning Research*, 17(1):1334–1373, 2016.

[73] Sergey Levine, Peter Pastor, Alex Krizhevsky, and Deirdre Quillen. Learning hand-eye coordination for robotic grasping with large-scale data collection. In *International Symposium on Experimental Robotics*, pages 173–184, 2016.

[74] Juan C Caicedo and Svetlana Lazebnik. Active object localization with deep reinforcement learning. In *IEEE Conference on International Conference on Computer Vision*, pages 2488–2496, 2015.

[75] Zequn Jie, Xiaodan Liang, Jiashi Feng, Xiaojie Jin, Wen Lu, and Shuicheng Yan. Tree-structured reinforcement learning for sequential object localization. In *Advances in Neural Information Processing Systems*, pages 127–135, 2016.

[76] Stefan Mathe, Aleksis Pirinen, and Cristian Sminchisescu. Reinforcement learning for visual object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2894–2902, 2016.

[77] Yansong Tang, Yi Tian, Jiwen Lu, Peiyang Li, and Jie Zhou. Deep progressive reinforcement learning for skeleton-based action recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5323–5332, 2018.

[78] Jianfu Zhang, Naiyan Wang, and Liqing Zhang. Multi-shot pedestrian re-identification via sequential decision making. *arXiv preprint arXiv:1712.07257*, 2017.

[79] Ronald J Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine Learning*, 8(3-4):229–256, 1992.

[80] David Silver, Guy Lever, Nicolas Heess, Thomas Degris, Daan Wierstra, and Martin A. Riedmiller. Deterministic policy gradient algorithms. In *Proceedings of the International Conference on Machine Learning*, volume 32, pages 387–395, 2014.

[81] Volodymyr Mnih, Adria Puigdomenech Badia, Mehdi Mirza, Alex Graves, Timothy Lillicrap, Tim Harley, David Silver, and Koray Kavukcuoglu. Asynchronous methods for deep reinforcement learning. In *International Conference on Machine Learning*, pages 1928–1937, 2016.

[82] Chen Huang, Simon Lucey, and Deva Ramanan. Learning policies for adaptive tracking with deep feature cascades. *arXiv preprint arXiv:1708.02973*, 2017.

[83] Sangdoo Yun, Jongwon Choi, Youngjoon Yoo, Kimin Yun, and Jin Young Choi. Action-decision networks for visual tracking with deep reinforcement learning. In

IEEE Conference on Computer Vision and Pattern Recognition, pages 1349–1358, 2017.

[84] Da Zhang, Hamid Maei, Xin Wang, and Yuan-Fang Wang. Deep reinforcement learning for visual object tracking in videos. *arXiv preprint arXiv:1701.08936*, 2017.

[85] James Steven Supancic III and Deva Ramanan. Tracking as online decision-making: Learning a policy from streaming videos with reinforcement learning. In *ICCV*, pages 322–331, 2017.

[86] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A Rusu, Joel Veness, Marc G Bellemare, Alex Graves, Martin Riedmiller, Andreas K Fidjeland, Georg Ostrovski, et al. Human-level control through deep reinforcement learning. *Nature*, 518(7540):529, 2015.

[87] David Silver, Aja Huang, Chris J Maddison, Arthur Guez, Laurent Sifre, George Van Den Driessche, Julian Schrittwieser, Ioannis Antonoglou, Veda Panneershelvam, Marc Lanctot, et al. Mastering the game of go with deep neural networks and tree search. *Nature*, 529(7587):484, 2016.

[88] Wenhan Luo, Peng Sun, Fangwei Zhong, Wei Liu, Tong Zhang, and Yizhou Wang. End-to-end active object tracking and its real-world deployment via reinforcement learning. *IEEE Trans. Pattern Anal. Mach. Intell.*, 42(6):1317–1332, 2020.

[89] S. Ren, K. He, R. Girshick, and J. Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(6):1137–1149, 2017.

[90] Raia Hadsell, Sumit Chopra, and Yann Lecun. Dimensionality reduction by learning an invariant mapping. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2006.

[91] Nitish Srivastava, Geoffrey E Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15(1):1929–1958, 2014.

[92] Ross Girshick. Fast r-cnn. In *IEEE Conference on International Conference on Computer Vision*, pages 1440–1448, 2015.

[93] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2016.

[94] Yangqing Jia, Evan Shelhamer, Jeff Donahue, Sergey Karayev, Jonathan Long, Ross Girshick, Sergio Guadarrama, and Trevor Darrell. Caffe: Convolutional architecture for fast feature embedding. In *Proceedings of the 22nd ACM International Conference on Multimedia*, pages 675–678, 2014.

[95] Piotr Dollár, Ron Appel, Serge Belongie, and Pietro Perona. Fast feature pyramids for object detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36(8):1532–1545, 2014.

[96] Bin Yang, Junjie Yan, Zhen Lei, and Stan Z Li. Convolutional channel features. In *IEEE Conference on International Conference on Computer Vision*, pages 82–90, 2015.

[97] Martin Koestinger, Martin Hirzer, Paul Wohlhart, Peter M Roth, and Horst Bischof. Large scale metric learning from equivalence constraints. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 2288–2295, 2012.

[98] Shengcai Liao, Yang Hu, Xiangyu Zhu, and Stan Z Li. Person re-identification by local maximal occurrence representation and metric learning. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 2197–2206, 2015.

[99] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In *International Conference on Learning Representations*, 2015.

[100] Sumit Chopra, Raia Hadsell, and Yann LeCun. Learning a similarity metric discriminatively, with application to face verification. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 539–546, 2005.

[101] Sam Hare, Stuart Golodetz, Amir Saffari, Vibhav Vineet, Ming-Ming Cheng, Stephen L. Hicks, and Philip H. S. Torr. Struck: Structured output tracking with kernels. *IEEE Trans. Pattern Anal. Mach. Intell.*, 38(10):2096–2109, 2016.

[102] Brox Thomas and Malik Jitendra. Large displacement optical flow: descriptor matching in variational motion estimation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(3):500–513, 2011.

[103] Naiyan Wang, Jianping Shi, Dit-Yan Yeung, and Jiaya Jia. Understanding and diagnosing visual tracking systems. In *IEEE International Conference on Computer Vision*, pages 3101–3109, 2015.

[104] Arnold W M Smeulders, Dung M Chu, Rita Cucchiara, Simone Calderara, Afshin Dehghan, and Mubarak Shah. Visual tracking: An experimental survey. In *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 1442–1468, 2014.

[105] Wei Zhong, Huchuan Lu, and Ming-Hsuan Yang. Robust object tracking via sparsity-based collaborative model. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1838–1845, 2012.

[106] Zhibin Hong, Zhe Chen, Chaohui Wang, Xue Mei, Danil V. Prokhorov, and Dacheng Tao. Multi-store tracker (muster): A cognitive psychology inspired approach to object tracking. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 749–758, 2015.

[107] Jin Gao, Haibin Ling, Weiming Hu, and Junliang Xing. Transfer learning based visual tracking with gaussian processes regression. In *European Conference on Computer Vision*, volume 8691, pages 188–203, 2014.

[108] Dafei Huang, Lei Luo, Mei Wen, Zhaoyun Chen, and Chunyuan Zhang. Enable scale and aspect ratio adaptability in visual tracking with detection proposals. In *European Conference on Computer Vision*, 2015.

[109] Jianming Zhang, Shugao Ma, and Stan Sclaroff. MEEM: robust tracking via multiple experts using entropy minimization. In *European Conference on Computer Vision*, volume 8694, pages 188–203, 2014.

[110] David S Bolme, J Ross Beveridge, Bruce A Draper, and Yui Man Lui. Visual object tracking using adaptive correlation filters. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 2544–2550, 2010.

[111] Yang Li and Jianke Zhu. A scale adaptive kernel correlation filter tracker with feature integration. In *European Conference on Computer Vision*, pages 254–265, 2014.

[112] Martin Danelljan, Gustav Hager, Fahad Shahbaz Khan, and Michael Felsberg. Adaptive decontamination of the training set: A unified formulation for discriminative visual tracking. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1430–1438, 2016.

[113] Susanna Gladh, Martin Danelljan, Fahad Shahbaz Khan, and Michael Felsberg. Deep motion features for visual tracking. In *International Conference on Pattern Recognition*, pages 1243–1248, 2016.

[114] Junyu Gao, Tianzhu Zhang, Xiaoshan Yang, and Changsheng Xu. Deep relative tracking. *IEEE Transactions on Image Processing*, 26(4):1845–1858, 2017.

[115] G. Zhu, F. Porikli, and H. Li. Not all negatives are equal: Learning to track with multiple background clusters. *IEEE Transactions on Circuits and Systems for Video Technology*, 28(2):314–326, 2018.

[116] W. Xue, C. Xu, and Z. Feng. Robust visual tracking via multi-scale spatio-temporal context learning. *IEEE Transactions on Circuits and Systems for Video Technology*, 28(10):2849–2860, 2018.

[117] Rui Yao, Guosheng Lin, Chunhua Shen, Yanning Zhang, and Qinfeng Shi. Semantics-aware visual object tracking. *IEEE Trans. Circuits Syst. Video Techn.*, 29(6):1687–1700, 2019.

[118] S. Wang, D. Wang, and H. Lu. Tracking with static and dynamic structured correlation filters. *IEEE Transactions on Circuits and Systems for Video Technology*, 28(10):2861–2869, 2018.

[119] Janghoon Choi, Junseok Kwon, and Kyoung Mu Lee. Real-time visual tracking by deep reinforced decision making. *Comput. Vis. Image Underst.*, 171:10–19, 2018.

[120] K. Chen and W. Tao. Once for all: A two-flow convolutional neural network for visual tracking. *IEEE Transactions on Circuits and Systems for Video Technology*, 28(12):3377–3386, 2018.

[121] Qing Guo, Wei Feng, Ce Zhou, Rui Huang, Liang Wan, and Song Wang. Learning dynamic siamese network for visual object tracking. In *Proc. IEEE Int. Conf. Comput. Vis.*, pages 1–9, 2017.

[122] Jack Valmadre, Luca Bertinetto, João Henriques, Andrea Vedaldi, and Philip HS Torr. End-to-end representation learning for correlation filter based tracking. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 5000–5008, 2017.

[123] Xingping Dong, Jianbing Shen, Wenguan Wang, Yu Liu, Ling Shao, and Fatih Porikli. Hyperparameter optimization for tracking with continuous deep q-learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 518–527, 2018.

[124] Po-Wei Chou, Daniel Maturana, and Sebastian Scherer. Improving stochastic policy gradients in continuous control with deep reinforcement learning using the beta

distribution. In *International Conference on Machine Learning*, pages 834–843, 2017.

[125] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision*, 115(3):211–252, 2015.

[126] Hailin Shi, Yang Yang, Xiangyu Zhu, Shengcai Liao, Zhen Lei, Weishi Zheng, and Stan Z Li. Embedding deep metric for person re-identification: A study against large variations. In *European Conference on Computer Vision*, pages 732–748, 2016.